This page is not part of

the document!

JP2005015085 / 2006-019143

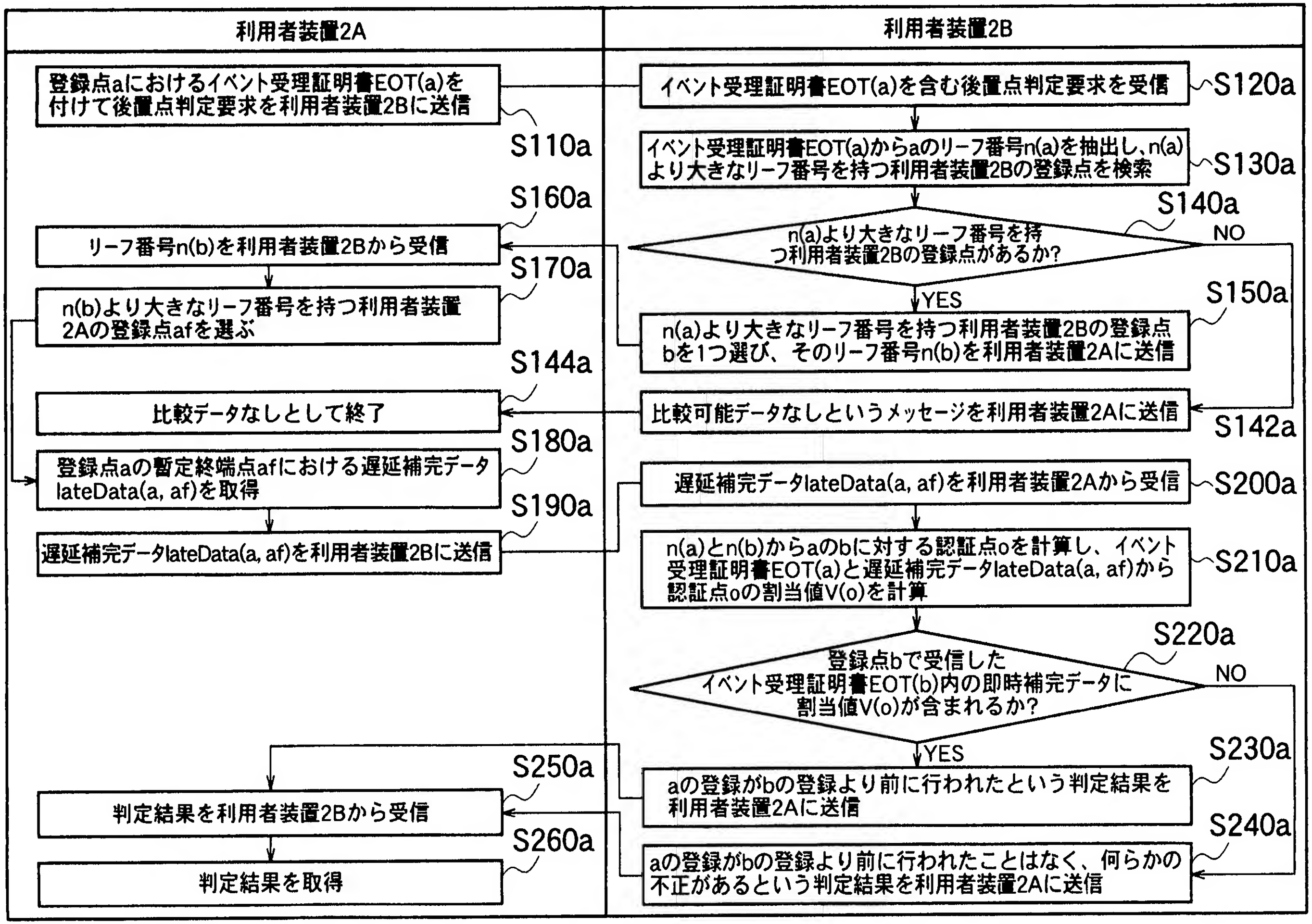
3/3

Date: Feb 23, 2006

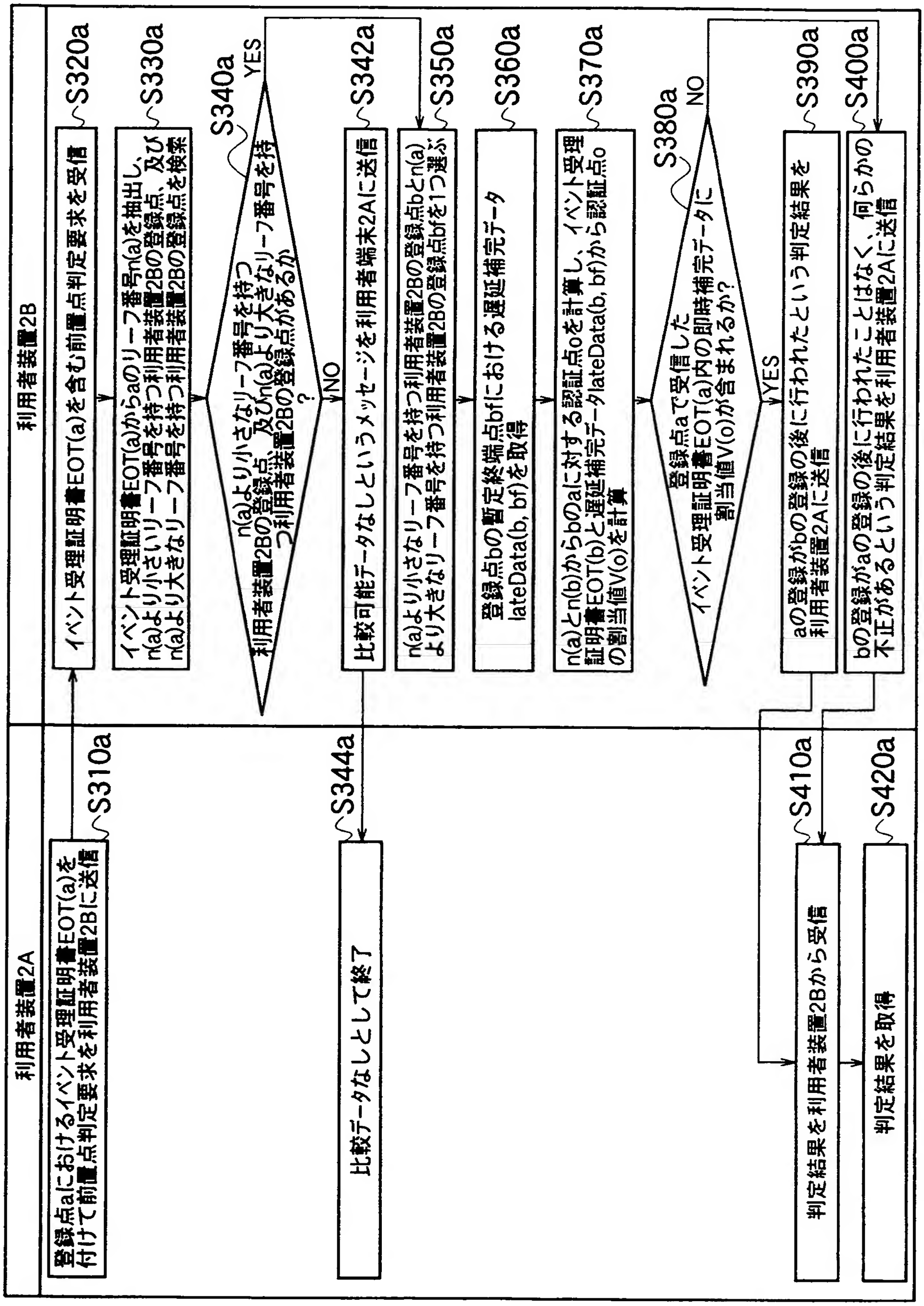
Recipient: IB

Address: Bureau international de l'OMPI 34, chemin des Colombettes Geneve 20 1211 CH

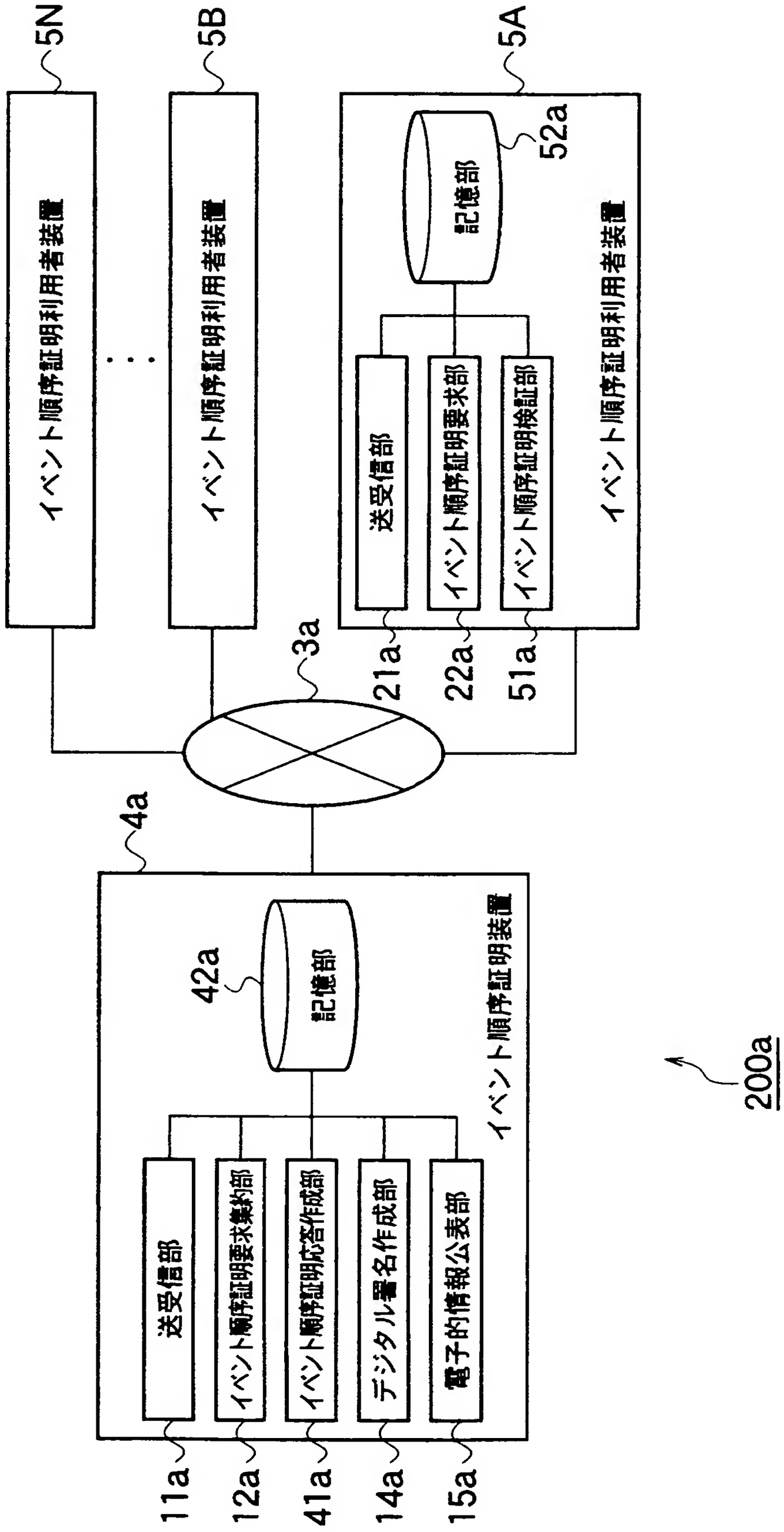
[図40]


利用者装置2A
利用者装置2B
登録点aにおけるイベント受理証明書EOT(a)を付けて後置点判定要求を利用者装置2Bに送信
S110a
イベント受理証明書EOT(a)を含む後置点判定要求を受信
S120a
イベント受理証明書EOT(a)からaのリーフ番号n(a)を抽出し、n(a)より大きなリーフ番号を持つ利用者装置2Bの登録点を検索
S130a
S140a
n(a)より大きなリーフ番号を持つ利用者装置2Bの登録点があるか?
NO
YES
S150a
n(a)より大きなリーフ番号を持つ利用者装置2Bの登録点bを1つ選び、そのリーフ番号n(b)を利用者装置2Aに送信
S160a
リーフ番号n(b)を利用者装置2Bから受信
S170a
n(b)より大きなリーフ番号を持つ利用者装置2Aの登録点afを選ぶ
S144a
比較データなしとして終了
比較可能データなしというメッセージを利用者装置2Aに送信
S142a
S180a
登録点aの暫定終端点afにおける遅延補完データlateData(a, af)を取得
S190a
遅延補完データlateData(a, af)を利用者装置2Bに送信
遅延補完データlateData(a, af)を利用者装置2Aから受信
S200a
n(a)とn(b)からaのbに対する認証点oを計算し、イベント受理証明書EOT(a)と遅延補完データlateData(a, af)から認証点oの割当値V(o)を計算
S210a
S220a
登録点bで受信したイベント受理証明書EOT(b)内の即時補完データに割当値V(o)が含まれるか?
NO
YES
S230a
aの登録がbの登録より前に行われたという判定結果を利用者装置2Aに送信
S240a
aの登録がbの登録より前に行われたことはなく、何らかの不正があるという判定結果を利用者装置2Aに送信
S250a
判定結果を利用者装置2Bから受信
S260a
判定結果を取得

[図41]


利用者装置2B
利用者装置2A
イベント受理証明書EOT(a)を含む前置点判定要求を受信
S320a
イベント受理証明書EOT(a)からaのリーフ番号n(a)を抽出し、
n(a)より小さいリーフ番号を持つ利用者装置2Bの登録点、及び
n(a)より大きなリーフ番号を持つ利用者装置2Bの登録点を検索
S330a
S340a
YES
n(a)より小さなリーフ番号を持つ利用者装置2Bの登録点、及びn(a)より大きなリーフ番号を持つ利用者装置2Bの登録点があるか？
NO
S342a
比較可能データなしというメッセージを利用者端末2Aに送信
S350a
n(a)より小さなリーフ番号を持つ利用者装置2Bの登録点bとn(a)より大きなリーフ番号を持つ利用者装置2Bの登録点bfを1つ選ぶ
S360a
登録点bの暫定終端点bfにおける遅延補完データlateData(b, bf)を取得
S370a
n(a)とn(b)からbのaに対する認証点oを計算し、イベント受理証明書EOT(b)と遅延補完データlateData(b, bf)から認証点oの割当値V(o)を計算
S380a
NO
登録点aで受信したイベント受理証明書EOT(a)内の即時補完データに割当値V(o)が含まれるか？
YES
S390a
aの登録がbの登録の後に行われたという判定結果を利用者装置2Aに送信
S400a
bの登録がaの登録の後に行われたことはなく、何らかの不正があるという判定結果を利用者装置2Aに送信
S310a
登録点aにおけるイベント受理証明書EOT(a)を付けて前置点判定要求を利用者装置2Bに送信
S344a
比較データなしとして終了
S410a
判定結果を利用者装置2Bから受信
S420a
判定結果を取得

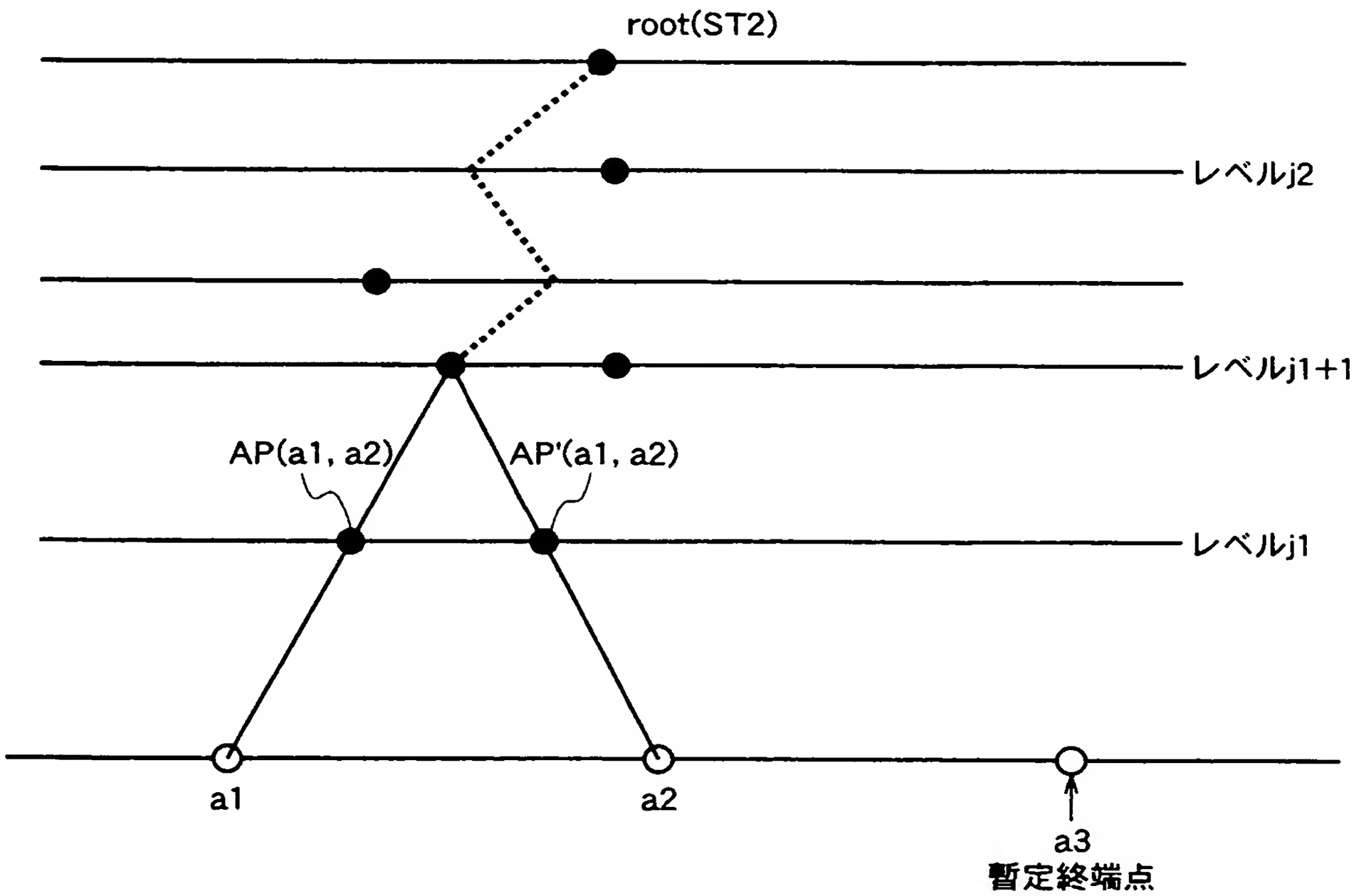
[図42]


5N
5B
5A
イベント順序証明利用者装置
イベント順序証明利用者装置
イベント順序証明利用者装置
記憶部
52a
送受信部
イベント順序証明要求部
イベント順序証明検証部
21a
22a
51a
3a
4a
記憶部
42a
イベント順序証明装置
送受信部
イベント順序証明要求集約部
イベント順序証明応答作成部
デジタル署名作成部
電子的情報公表部
11a
12a
41a
14a
15a
200a

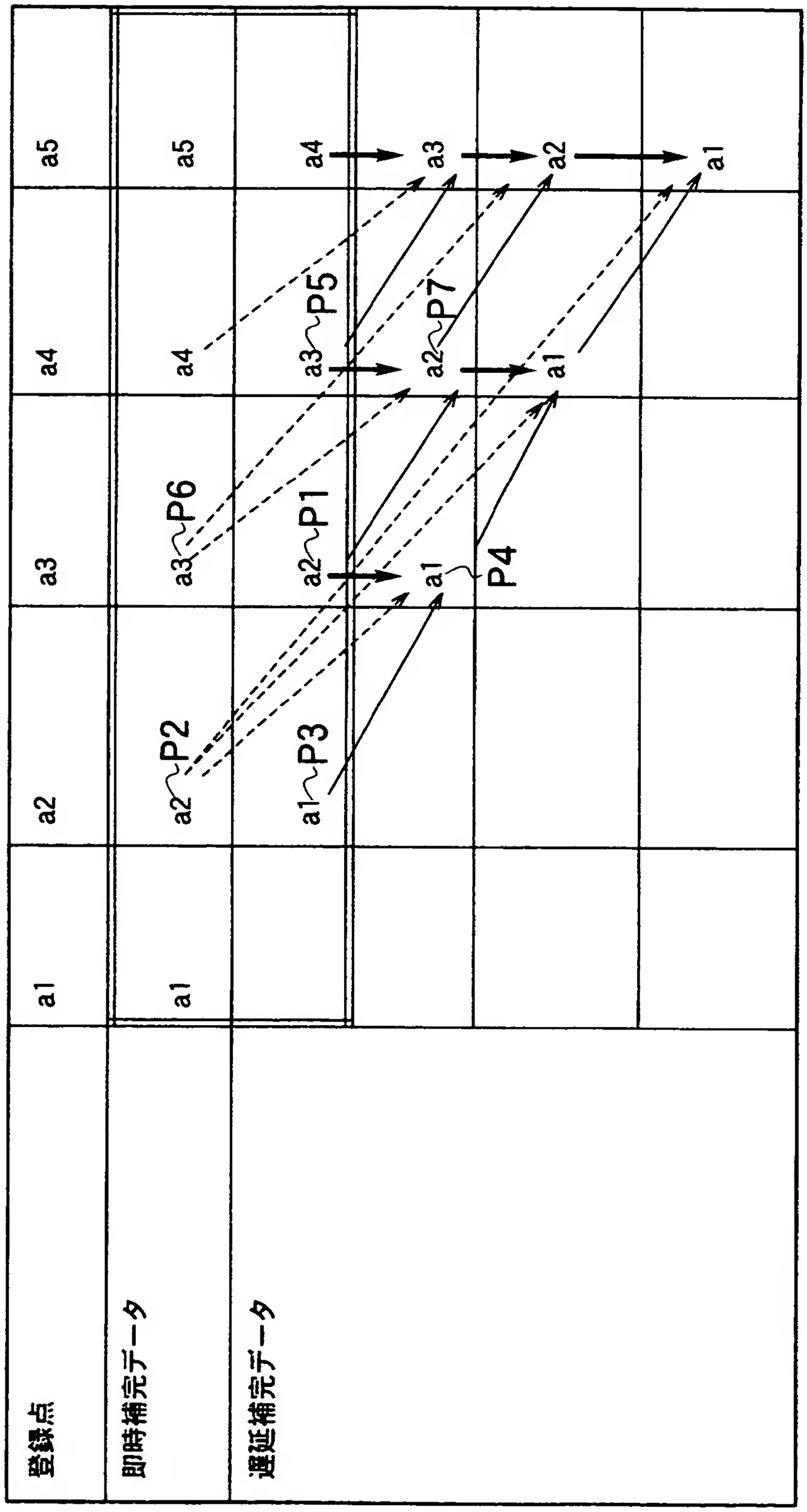
[図43]

項目	記号	必須	
元デジタルデータ	y	○	イベント順序受理証明書 EOC(y)
順次割当データ	z	○	
順次集約木識別番号	n	○	
順次集約木リーフ番号	i	○	
登録点の即時補完データ(位置情報、割当値)	SK	○	
直前登録点の遅延補完データ(位置情報、割当値)	TK2	○	

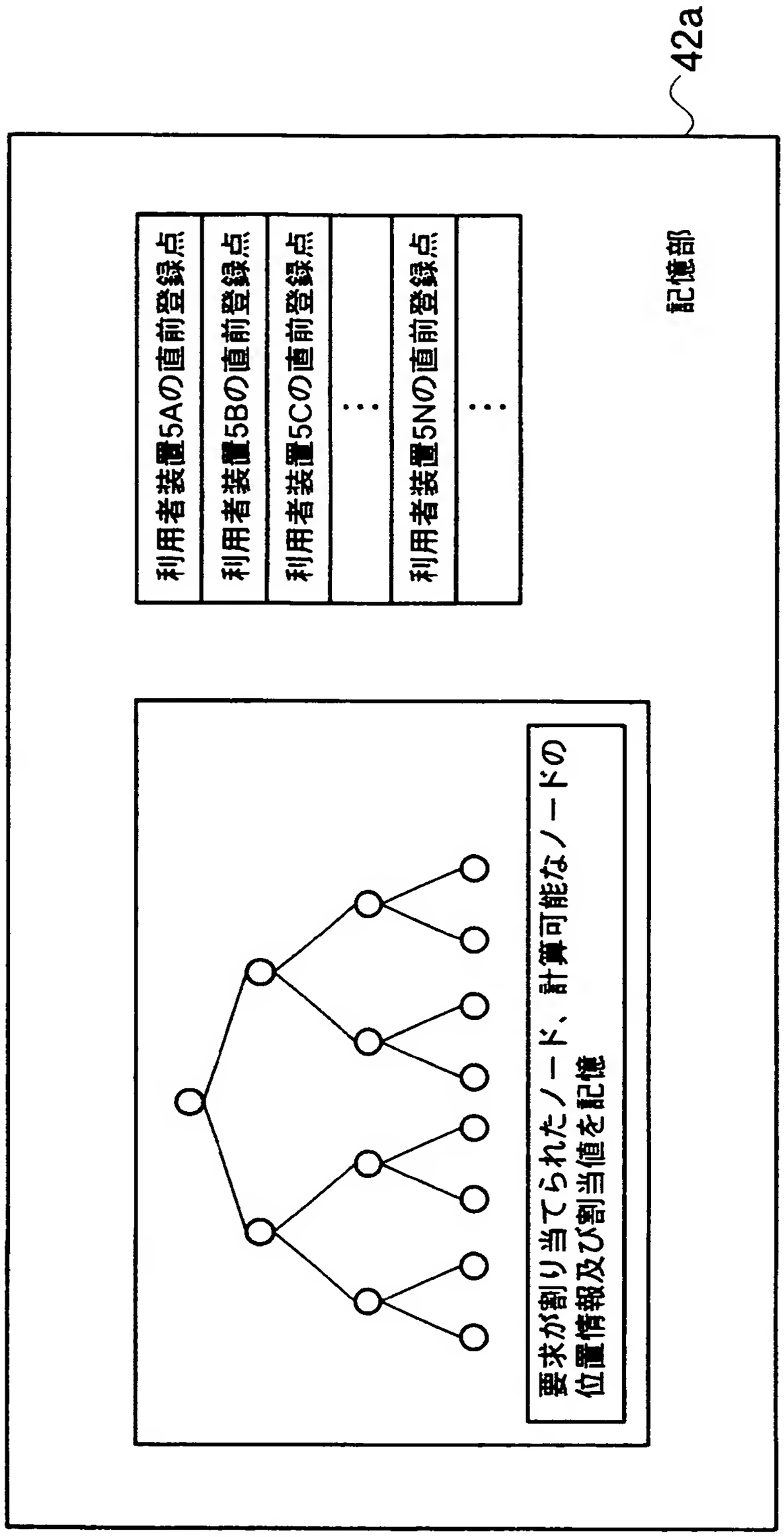
[図44]



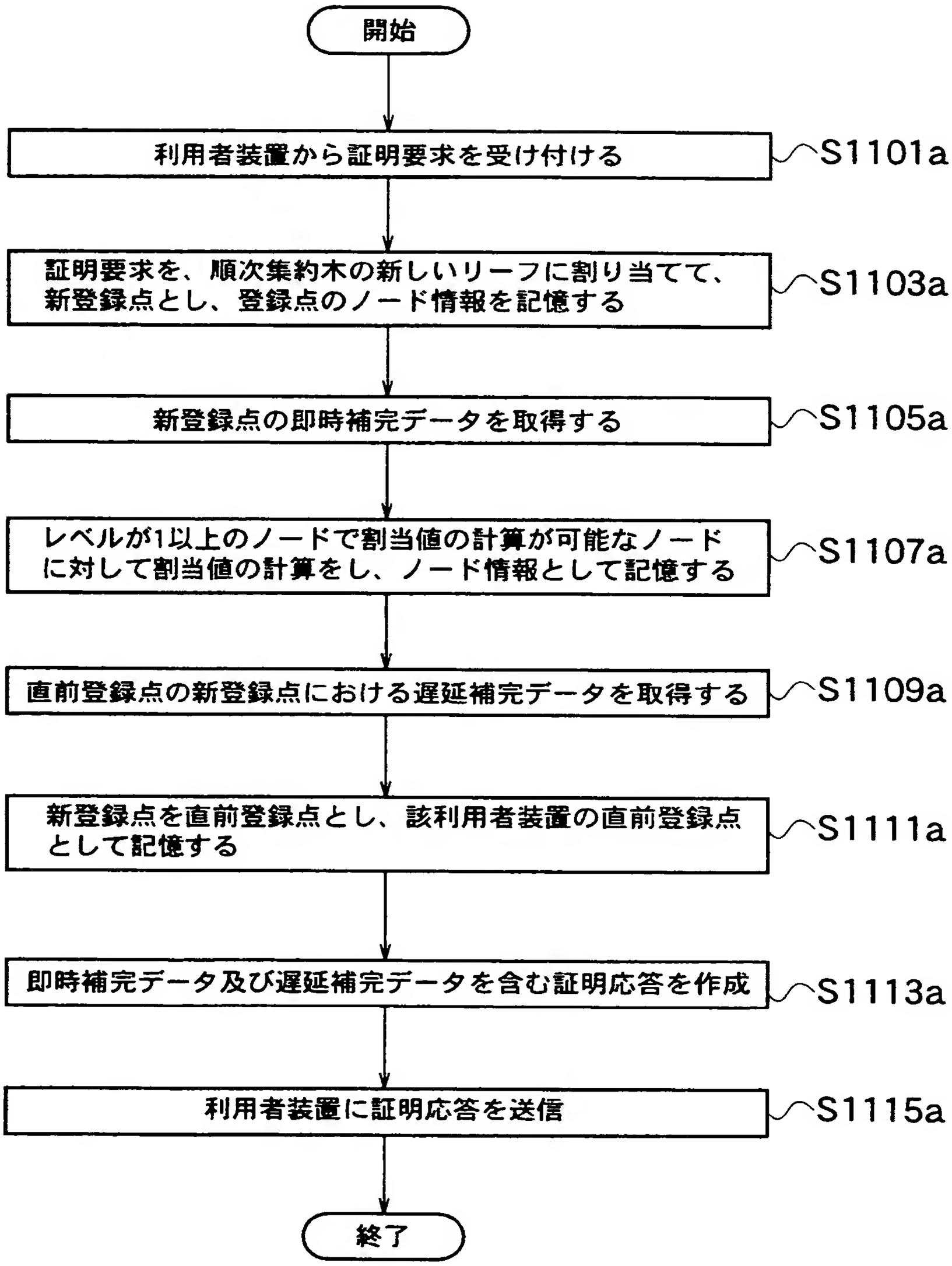
[図45]


a5
a5
a4
a3
a2
a1
a4
a4
a3
P5
a2
P7
a1
a3
a3
P6
a2
P1
a1
P4
a2
a2
P2
a1
P3
a1
a1
登録点
即時補完データ
遅延補完データ

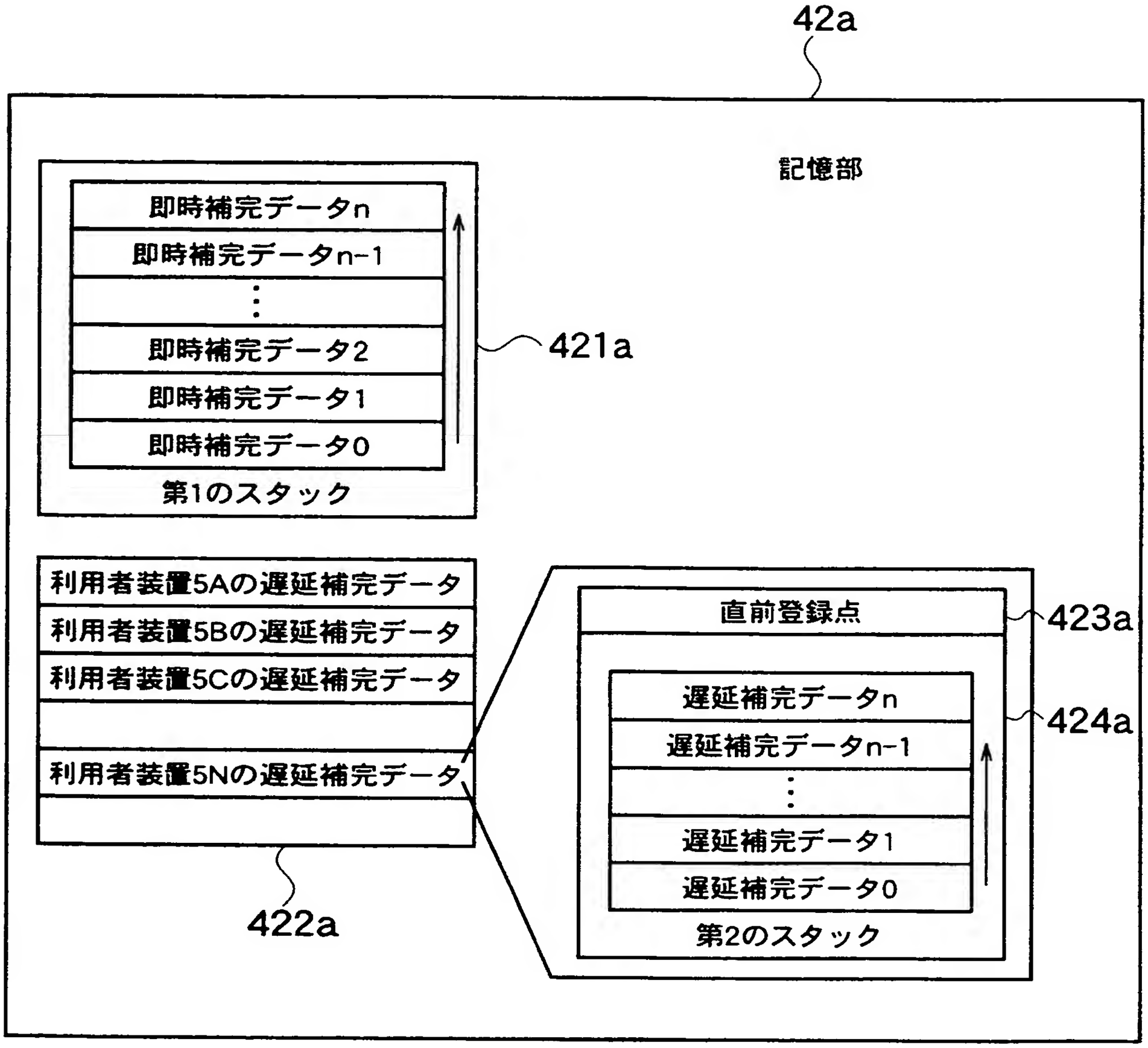
[図46]


42a
記憶部
利用者装置5Aの直前登録点
利用者装置5Bの直前登録点
利用者装置5Cの直前登録点
利用者装置5Nの直前登録点
要求が割り当てられたノード、計算可能なノードの
位置情報及び割当値を記憶

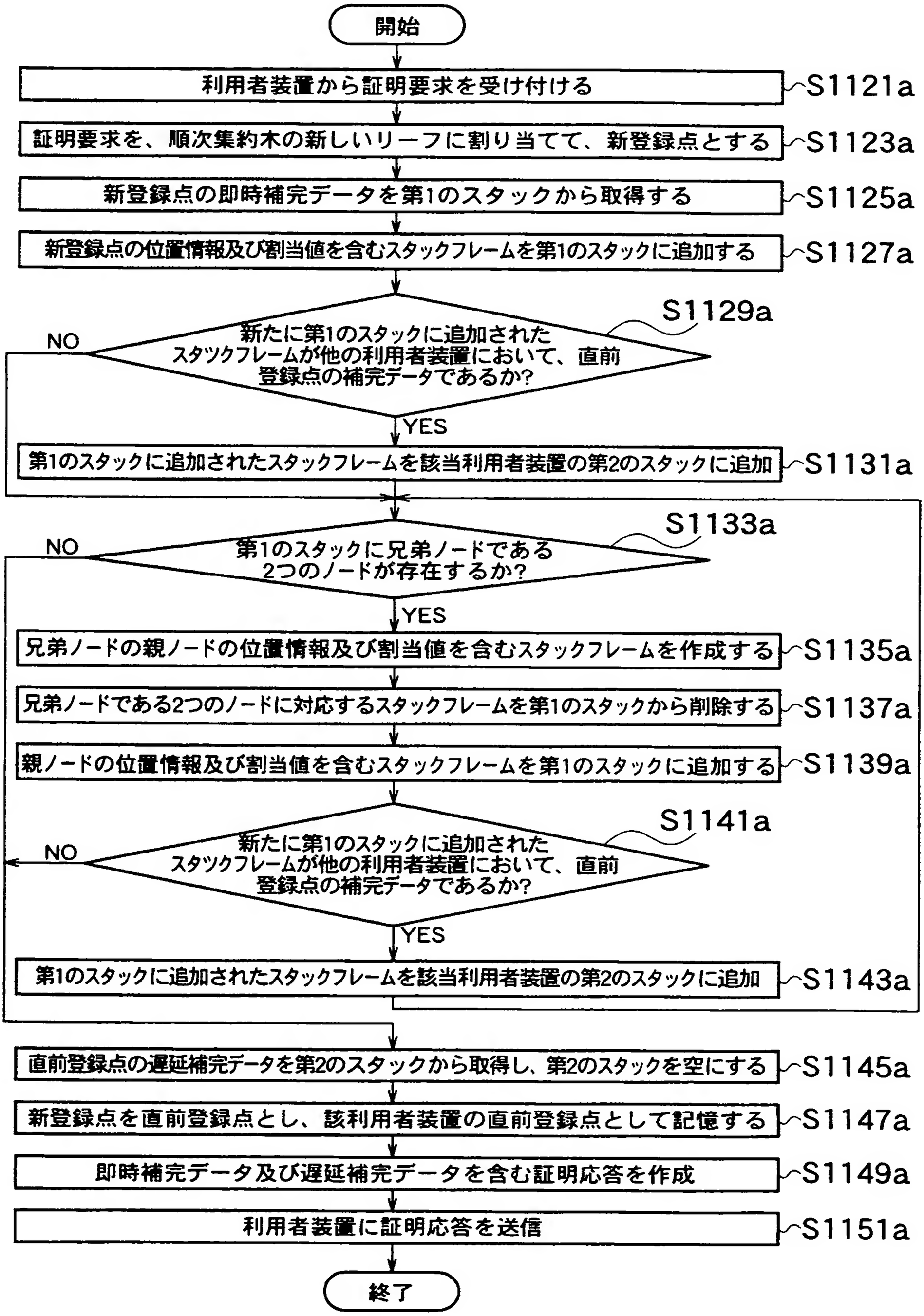
[図47]


開始
利用者装置から証明要求を受け付ける
S1101a
証明要求を、順次集約木の新しいリーフに割り当てて、新登録点とし、登録点のノード情報を記憶する
S1103a
新登録点の即時補完データを取得する
S1105a
レベルが1以上のノードで割当値の計算が可能なノードに対して割当値の計算をし、ノード情報として記憶する
S1107a
直前登録点の新登録点における遅延補完データを取得する
S1109a
新登録点を直前登録点とし、該利用者装置の直前登録点として記憶する
S1111a
即時補完データ及び遅延補完データを含む証明応答を作成
S1113a
利用者装置に証明応答を送信
S1115a
終了

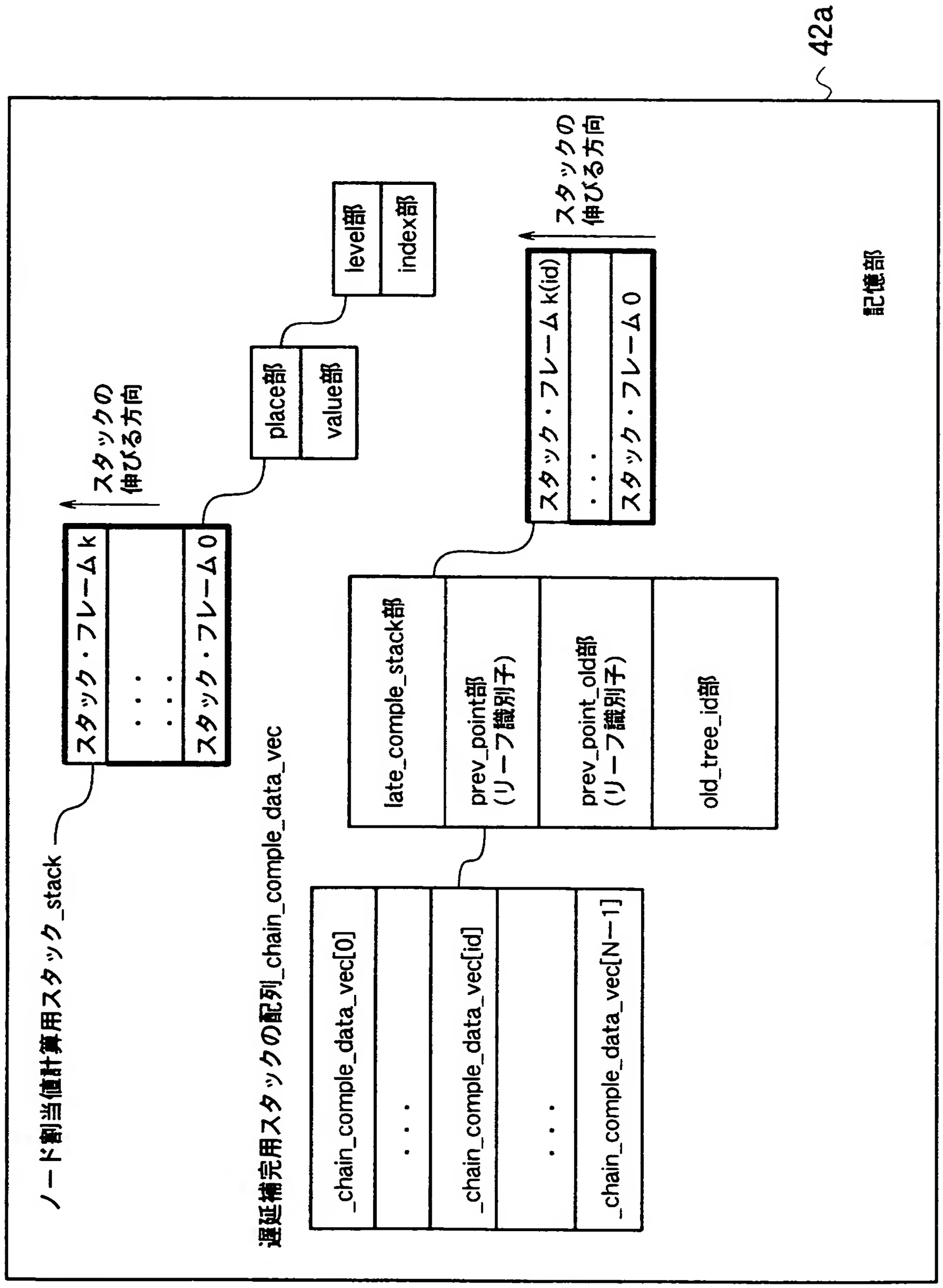
[図48]


42a
記憶部
即時補完データn
即時補完データn-1
即時補完データ2
即時補完データ1
即時補完データ0
第1のスタック
421a
利用者装置5Aの遅延補完データ
利用者装置5Bの遅延補完データ
利用者装置5Cの遅延補完データ
利用者装置5Nの遅延補完データ
422a
直前登録点
423a
遅延補完データn
遅延補完データn-1
遅延補完データ1
遅延補完データ0
第2のスタック
424a

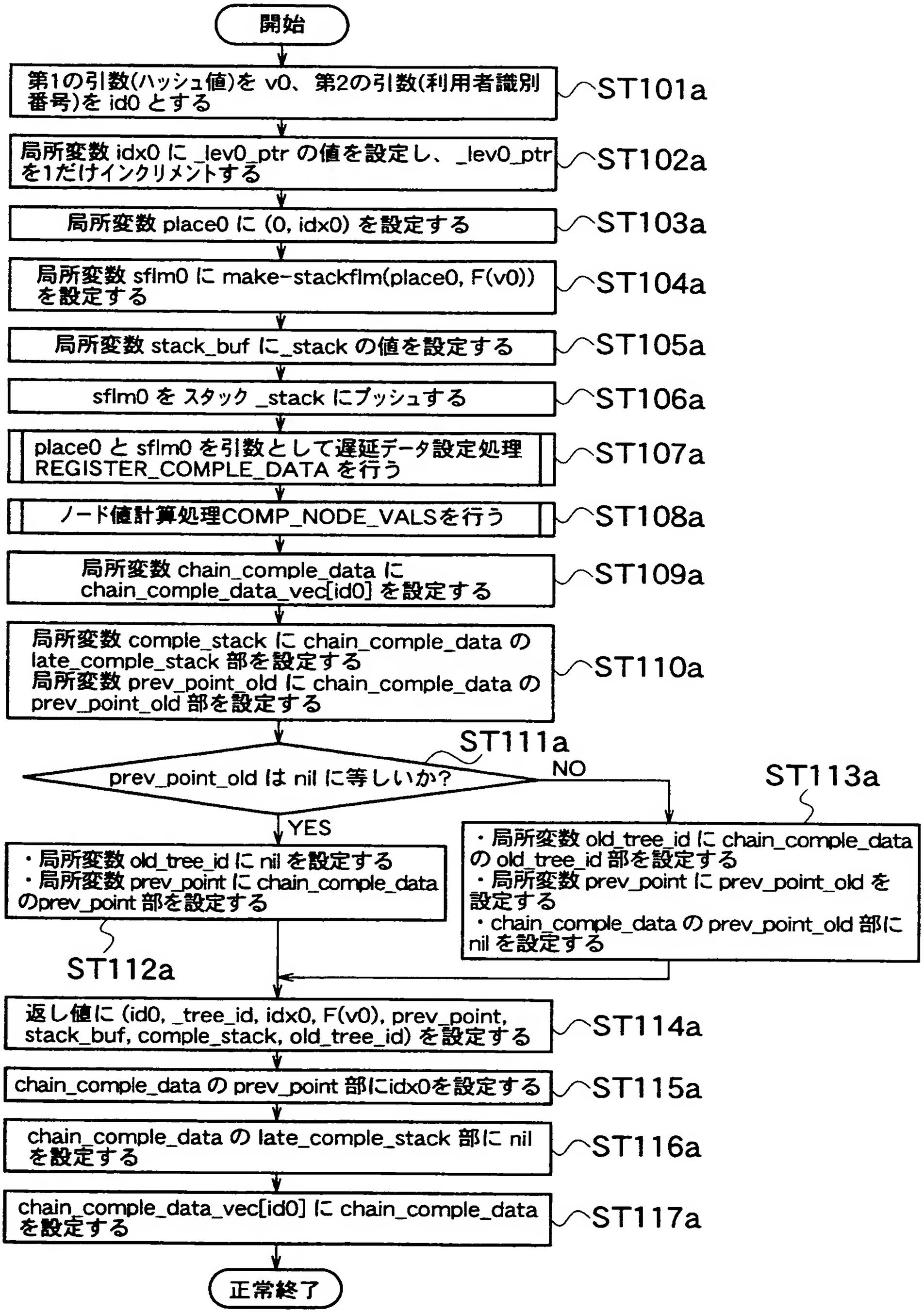
[図49]


開始
利用者装置から証明要求を受け付ける
S1121a
証明要求を、順次集約木の新しいリーフに割り当てて、新登録点とする
S1123a
新登録点の即時補完データを第1のスタックから取得する
S1125a
新登録点の位置情報及び割当値を含むスタックフレームを第1のスタックに追加する
S1127a
S1129a
新たに第1のスタックに追加された
スタックフレームが他の利用者装置において、直前
登録点の補完データであるか?
NO
YES
第1のスタックに追加されたスタックフレームを該当利用者装置の第2のスタックに追加
S1131a
S1133a
第1のスタックに兄弟ノードである
2つのノードが存在するか?
NO
YES
兄弟ノードの親ノードの位置情報及び割当値を含むスタックフレームを作成する
S1135a
兄弟ノードである2つのノードに対応するスタックフレームを第1のスタックから削除する
S1137a
親ノードの位置情報及び割当値を含むスタックフレームを第1のスタックに追加する
S1139a
S1141a
新たに第1のスタックに追加された
スタックフレームが他の利用者装置において、直前
登録点の補完データであるか?
NO
YES
第1のスタックに追加されたスタックフレームを該当利用者装置の第2のスタックに追加
S1143a
直前登録点の遅延補完データを第2のスタックから取得し、第2のスタックを空にする
S1145a
新登録点を直前登録点とし、該利用者装置の直前登録点として記憶する
S1147a
即時補完データ及び遅延補完データを含む証明応答を作成
S1149a
利用者装置に証明応答を送信
S1151a
終了

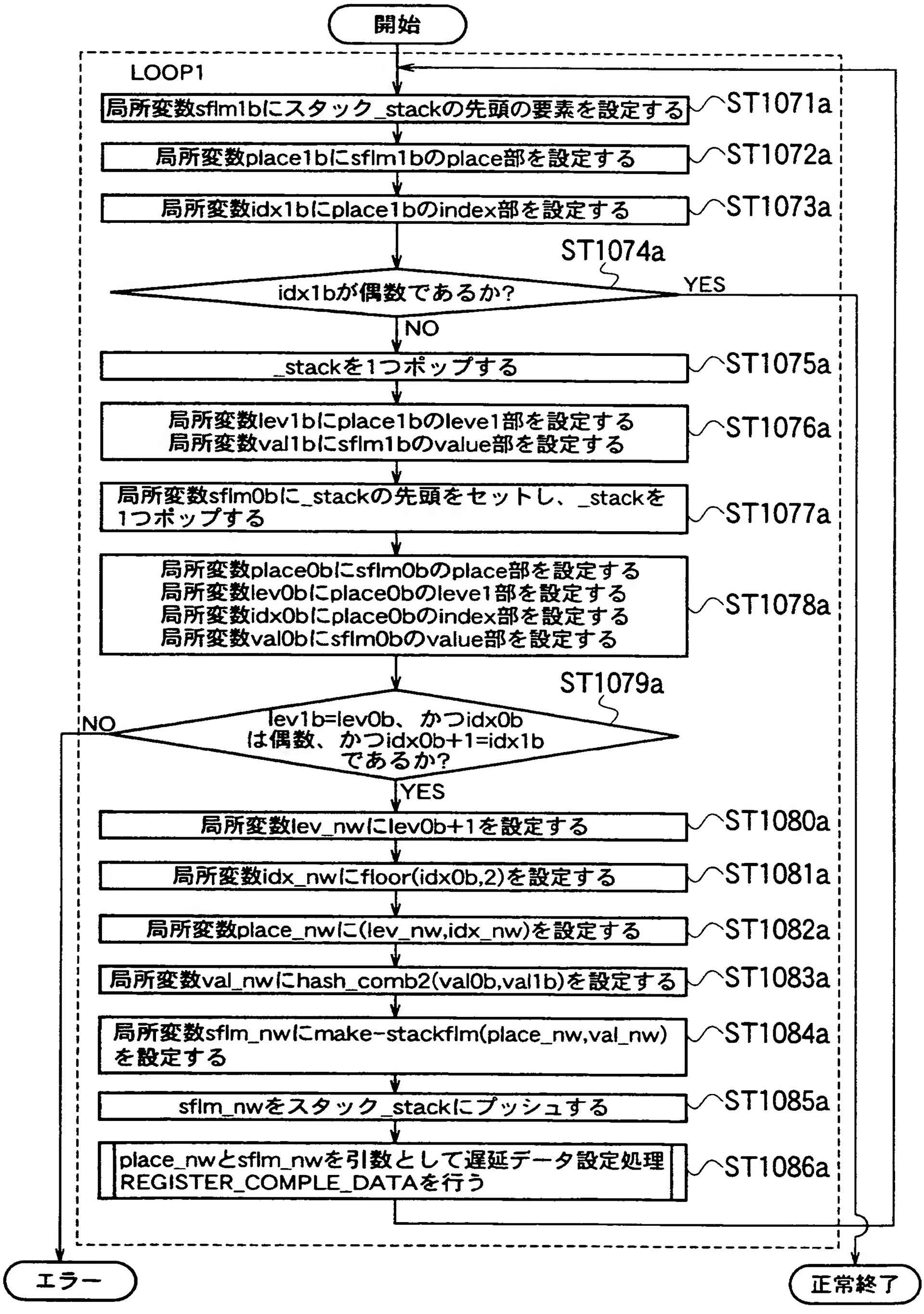
[図50]


42a
記憶部
スタックの伸びる方向
level部
index部
place部
value部
スタック・フレーム k
スタック・フレーム 0
ノード割当値計算用スタック _stack
late_comple_stack部
prev_point部
(リーフ識別子)
prev_point_old部
(リーフ識別子)
old_tree_id部
遅延補完用スタックの配列 _chain_comple_data_vec
_chain_comple_data_vec[0]
_chain_comple_data_vec[id]
_chain_comple_data_vec[N−1]

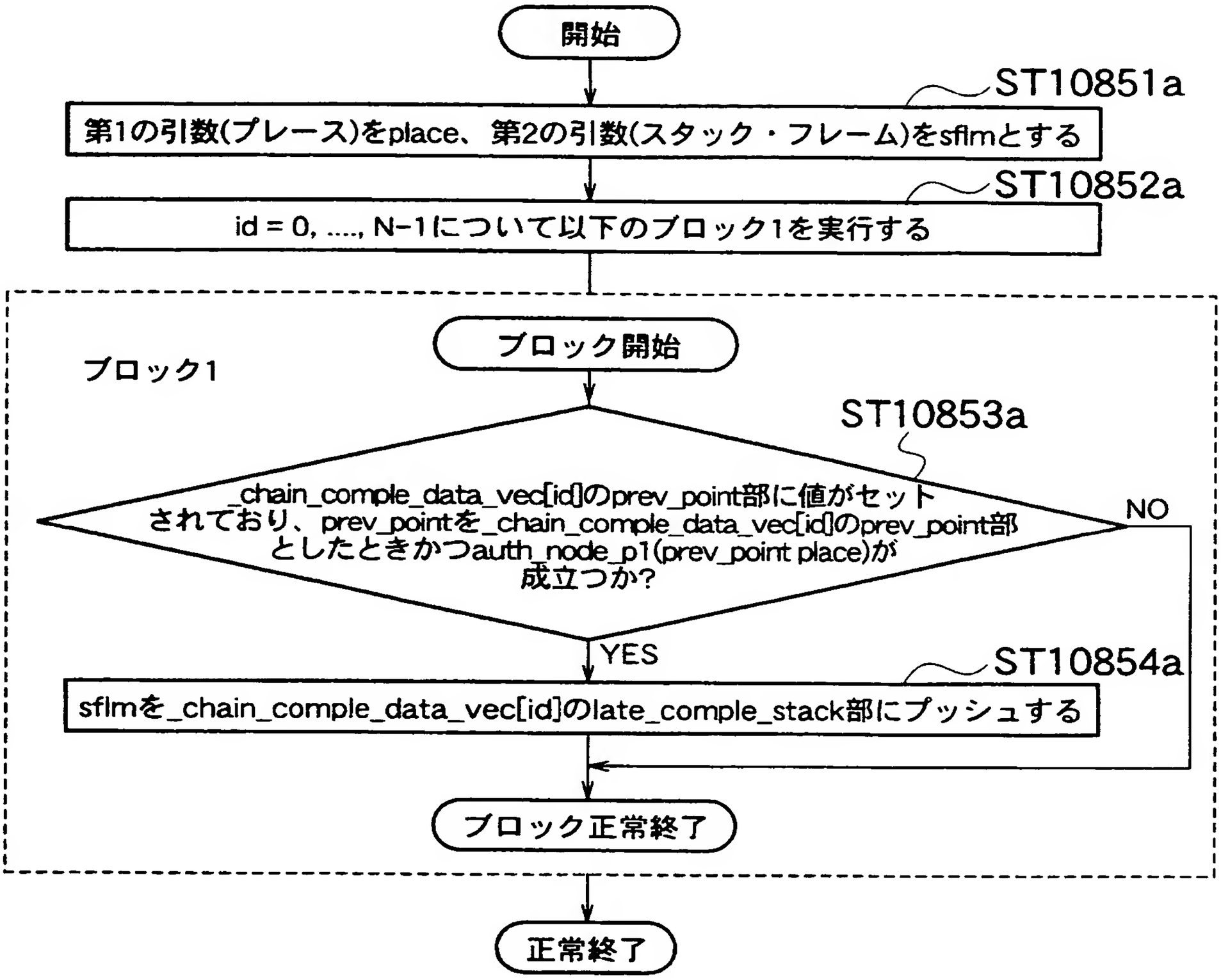
[図51]


開始
第1の引数(ハッシュ値)を v0、第2の引数(利用者識別番号)を id0 とする
ST101a
局所変数 idx0 に _lev0_ptr の値を設定し、_lev0_ptr を1だけインクリメントする
ST102a
局所変数 place0 に (0, idx0) を設定する
ST103a
局所変数 sflm0 に make-stackflm(place0, F(v0)) を設定する
ST104a
局所変数 stack_buf に_stack の値を設定する
ST105a
sflm0 を スタック _stack にプッシュする
ST106a
place0 と sflm0 を引数として遅延データ設定処理 REGISTER_COMPLE_DATA を行う
ST107a
ノード値計算処理COMP_NODE_VALSを行う
ST108a
局所変数 chain_comple_data に chain_comple_data_vec[id0] を設定する
ST109a
局所変数 comple_stack に chain_comple_data の late_comple_stack 部を設定する
局所変数 prev_point_old に chain_comple_data の prev_point_old 部を設定する
ST110a
ST111a
prev_point_old は nil に等しいか?
NO
YES
ST113a
・局所変数 old_tree_id に chain_comple_data の old_tree_id 部を設定する
・局所変数 prev_point に prev_point_old を設定する
・chain_comple_data の prev_point_old 部に nil を設定する
・局所変数 old_tree_id に nil を設定する
・局所変数 prev_point に chain_comple_data のprev_point 部を設定する
ST112a
返し値に (id0, _tree_id, idx0, F(v0), prev_point, stack_buf, comple_stack, old_tree_id) を設定する
ST114a
chain_comple_data の prev_point 部にidx0を設定する
ST115a
chain_comple_data の late_comple_stack 部に nil を設定する
ST116a
chain_comple_data_vec[id0] に chain_comple_data を設定する
ST117a
正常終了

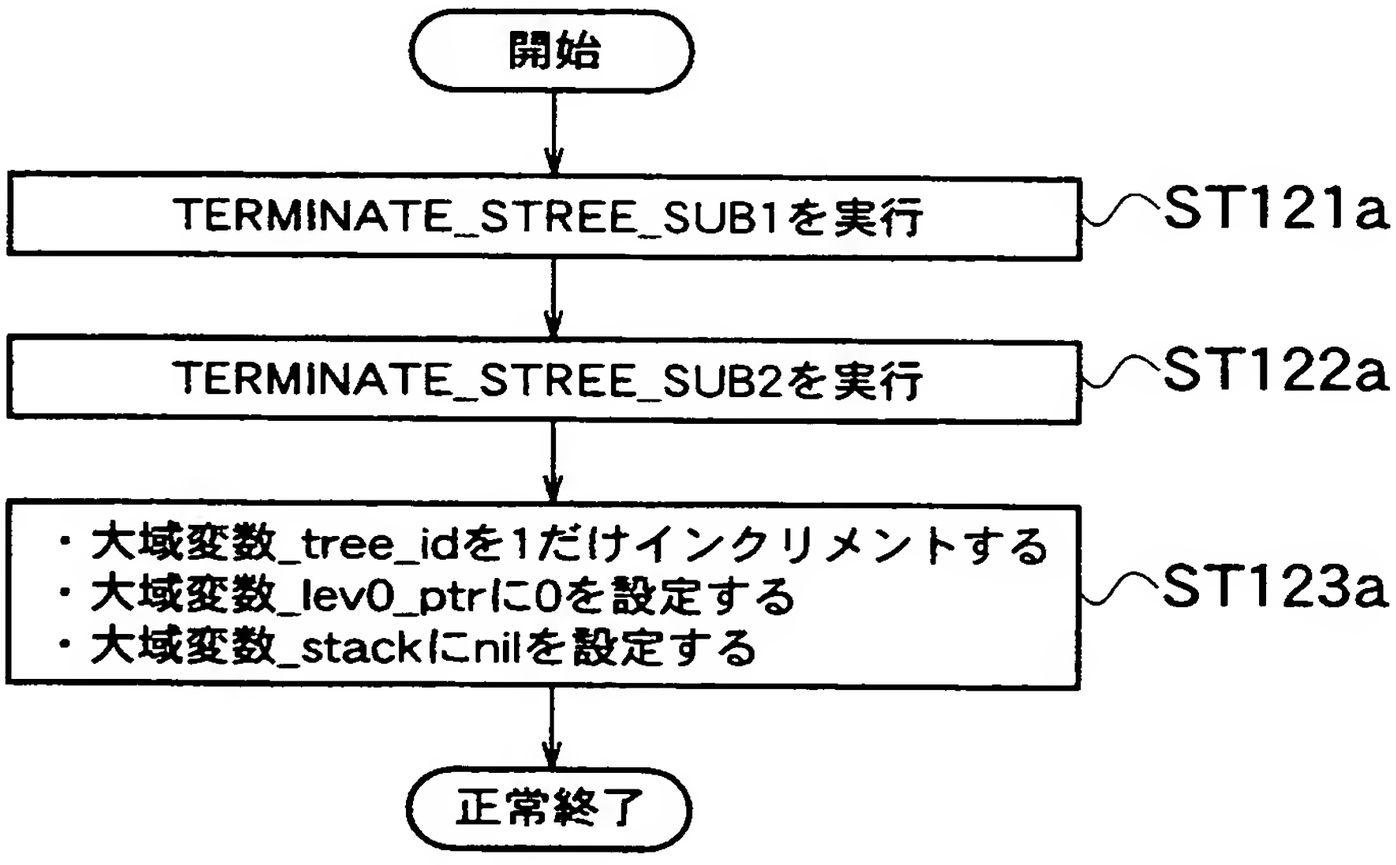
[図52]


開始
LOOP1
局所変数sflm1bにスタック_stackの先頭の要素を設定する
ST1071a
局所変数place1bにsflm1bのplace部を設定する
ST1072a
局所変数idx1bにplace1bのindex部を設定する
ST1073a
ST1074a
idx1bが偶数であるか?
YES
NO
_stackを1つポップする
ST1075a
局所変数lev1bにplace1bのleve1部を設定する
局所変数val1bにsflm1bのvalue部を設定する
ST1076a
局所変数sflm0bに_stackの先頭をセットし、_stackを1つポップする
ST1077a
局所変数place0bにsflm0bのplace部を設定する
局所変数lev0bにplace0bのleve1部を設定する
局所変数idx0bにplace0bのindex部を設定する
局所変数val0bにsflm0bのvalue部を設定する
ST1078a
ST1079a
lev1b=lev0b、かつidx0b
は偶数、かつidx0b+1=idx1b
であるか?
NO
YES
局所変数lev_nwにlev0b+1を設定する
ST1080a
局所変数idx_nwにfloor(idx0b,2)を設定する
ST1081a
局所変数place_nwに(lev_nw,idx_nw)を設定する
ST1082a
局所変数val_nwにhash_comb2(val0b,val1b)を設定する
ST1083a
局所変数sflm_nwにmake-stackflm(place_nw,val_nw)を設定する
ST1084a
sflm_nwをスタック_stackにプッシュする
ST1085a
place_nwとsflm_nwを引数として遅延データ設定処理REGISTER_COMPLE_DATAを行う
ST1086a
エラー
正常終了

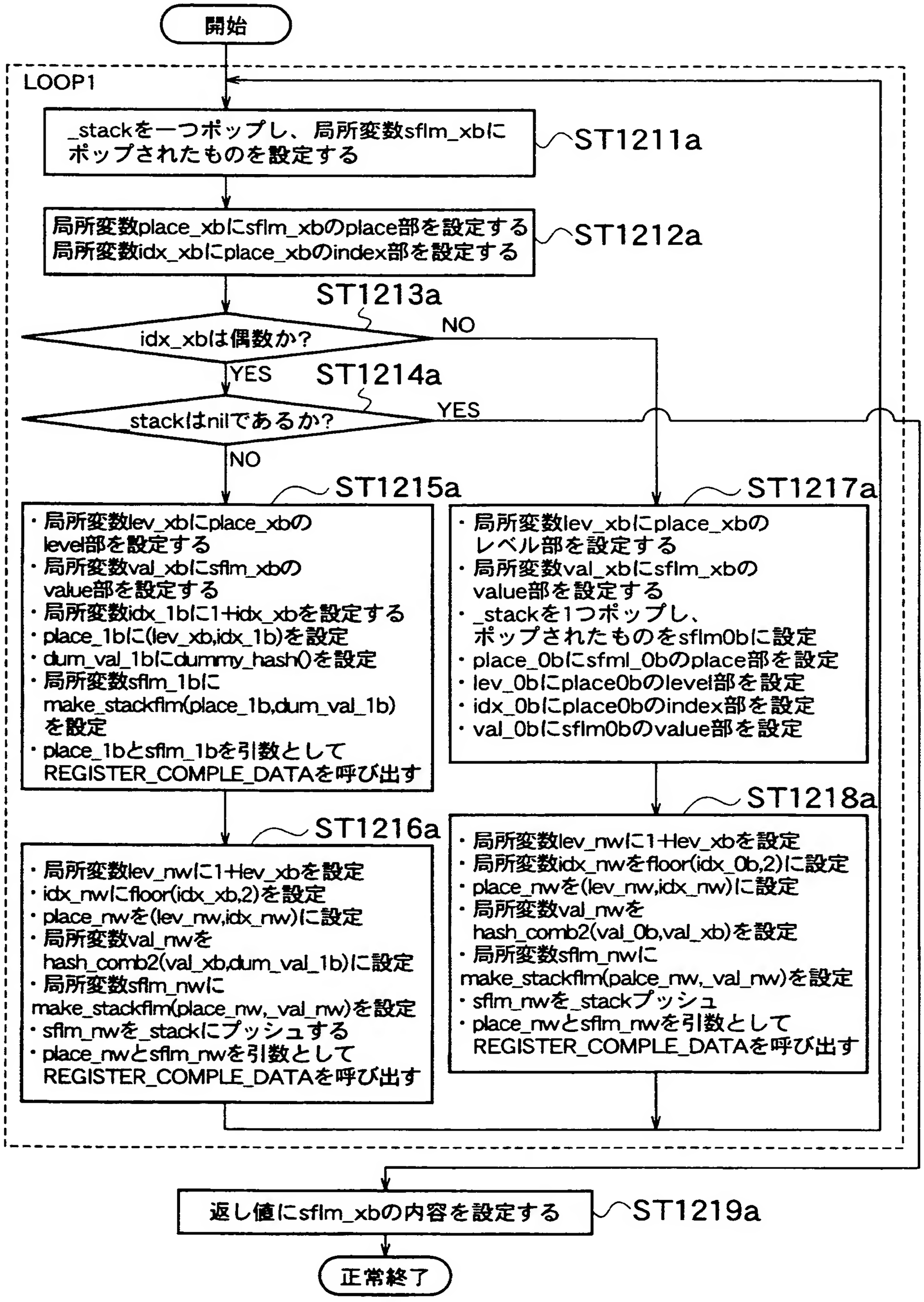
[図53]


開始
ST10851a
第1の引数(プレース)をplace、第2の引数(スタック・フレーム)をsflmとする
ST10852a
id = 0, ..., N-1について以下のブロック1を実行する
ブロック1
ブロック開始
ST10853a
_chain_comple_data_vec[id]のprev_point部に値がセットされており、prev_pointを_chain_comple_data_vec[id]のprev_point部としたときかつauth_node_p1(prev_point place)が成立つか?
NO
YES
ST10854a
sflmを_chain_comple_data_vec[id]のlate_comple_stack部にプッシュする
ブロック正常終了
正常終了

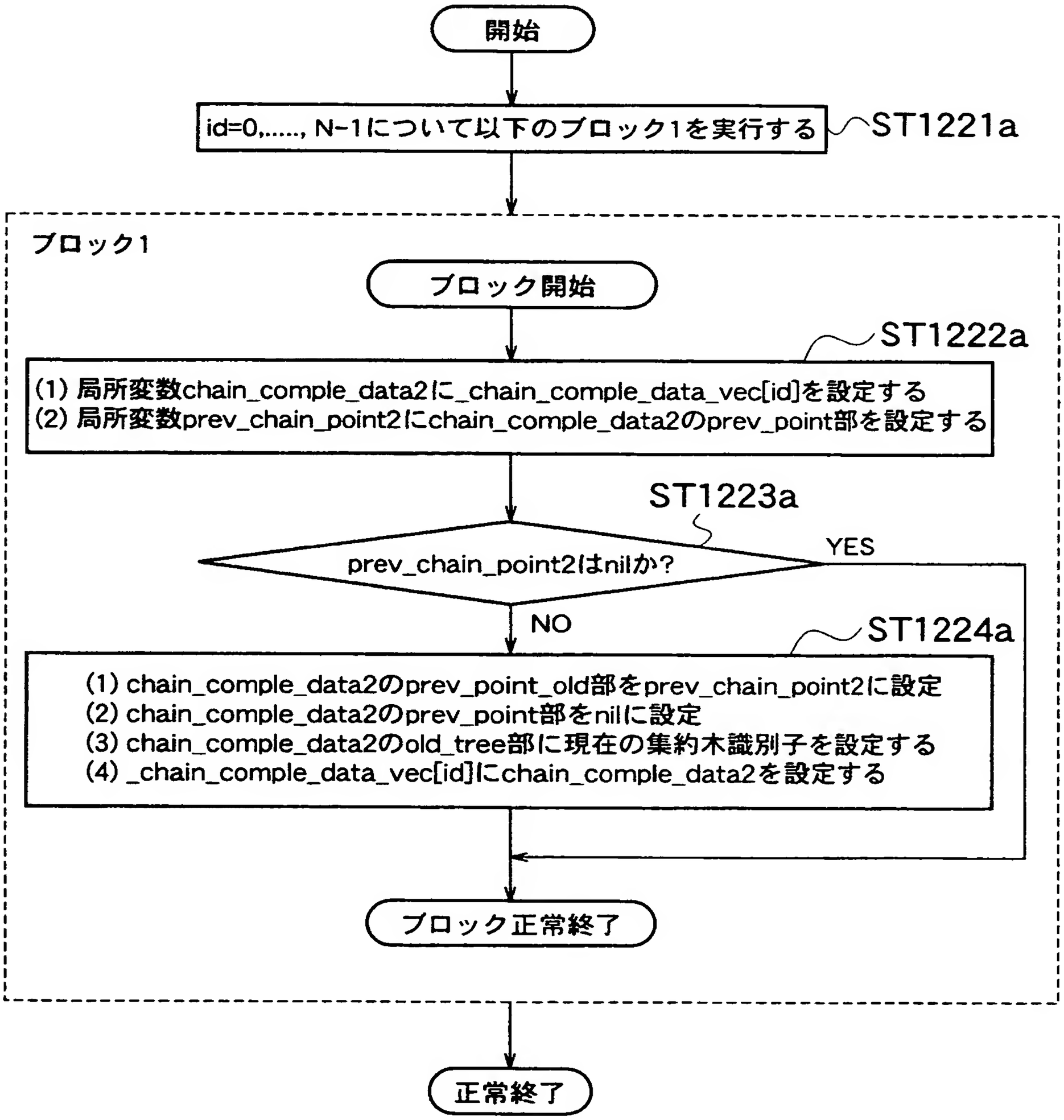
[図54]


開始
TERMINATE_STREE_SUB1を実行
ST121a
TERMINATE_STREE_SUB2を実行
ST122a
・大域変数_tree_idを1だけインクリメントする
・大域変数_lev0_ptrに0を設定する
・大域変数_stackにnilを設定する
ST123a
正常終了

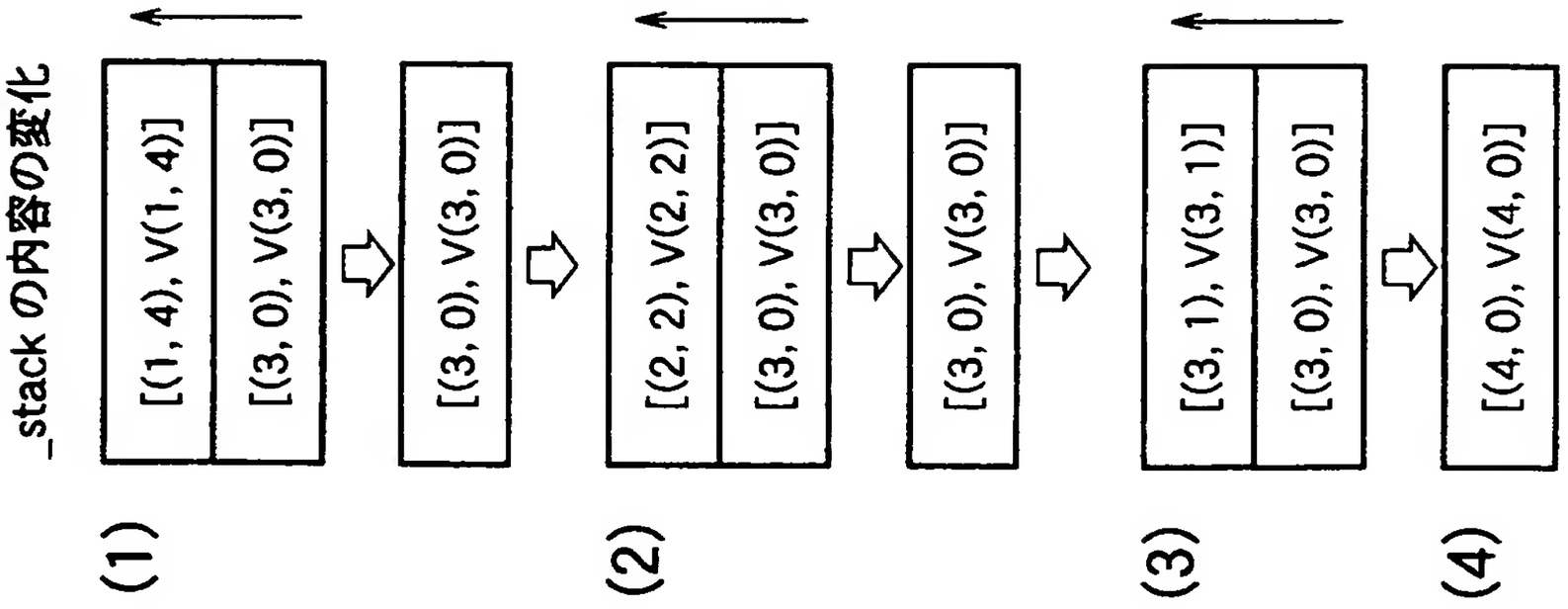
[図55]

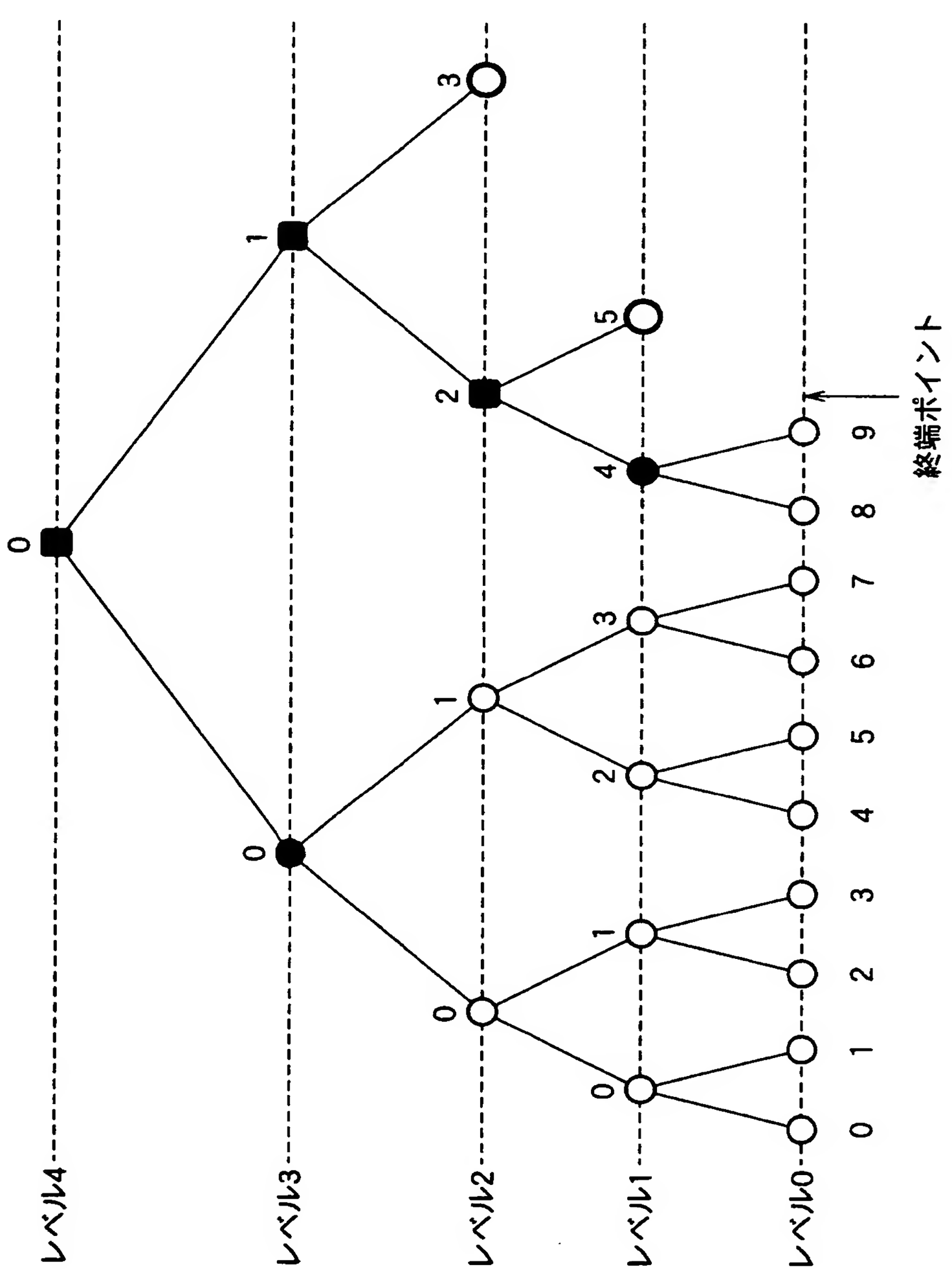

開始
LOOP1
_stackを一つポップし、局所変数sflm_xbにポップされたものを設定する
ST1211a
局所変数place_xbにsflm_xbのplace部を設定する
局所変数idx_xbにplace_xbのindex部を設定する
ST1212a
ST1213a
idx_xbは偶数か?
NO
YES
ST1214a
stackはnilであるか?
YES
NO
ST1215a
・局所変数lev_xbにplace_xbのlevel部を設定する
・局所変数val_xbにsflm_xbのvalue部を設定する
・局所変数idx_1bに1+idx_xbを設定する
・place_1bに(lev_xb,idx_1b)を設定
・dum_val_1bにdummy_hash()を設定
・局所変数sflm_1bにmake_stackflm(place_1b,dum_val_1b)を設定
・place_1bとsflm_1bを引数としてREGISTER_COMPLE_DATAを呼び出す
ST1217a
・局所変数lev_xbにplace_xbのレベル部を設定する
・局所変数val_xbにsflm_xbのvalue部を設定する
・_stackを1つポップし、ポップされたものをsflm0bに設定
・place_0bにsfml_0bのplace部を設定
・lev_0bにplace0bのlevel部を設定
・idx_0bにplace0bのindex部を設定
・val_0bにsflm0bのvalue部を設定
ST1218a
ST1216a
・局所変数lev_nwに1+lev_xbを設定
・idx_nwにfloor(idx_xb,2)を設定
・place_nwを(lev_nw,idx_nw)に設定
・局所変数val_nwをhash_comb2(val_xb,dum_val_1b)に設定
・局所変数sflm_nwにmake_stackflm(place_nw,_val_nw)を設定
・sflm_nwを_stackにプッシュする
・place_nwとsflm_nwを引数としてREGISTER_COMPLE_DATAを呼び出す
・局所変数lev_nwに1+lev_xbを設定
・局所変数idx_nwをfloor(idx_0b,2)に設定
・place_nwを(lev_nw,idx_nw)に設定
・局所変数val_nwをhash_comb2(val_0b,val_xb)を設定
・局所変数sflm_nwにmake_stackflm(palce_nw,_val_nw)を設定
・sflm_nwを_stackプッシュ
・place_nwとsflm_nwを引数としてREGISTER_COMPLE_DATAを呼び出す
返し値にsflm_xbの内容を設定する
ST1219a
正常終了

[図56]

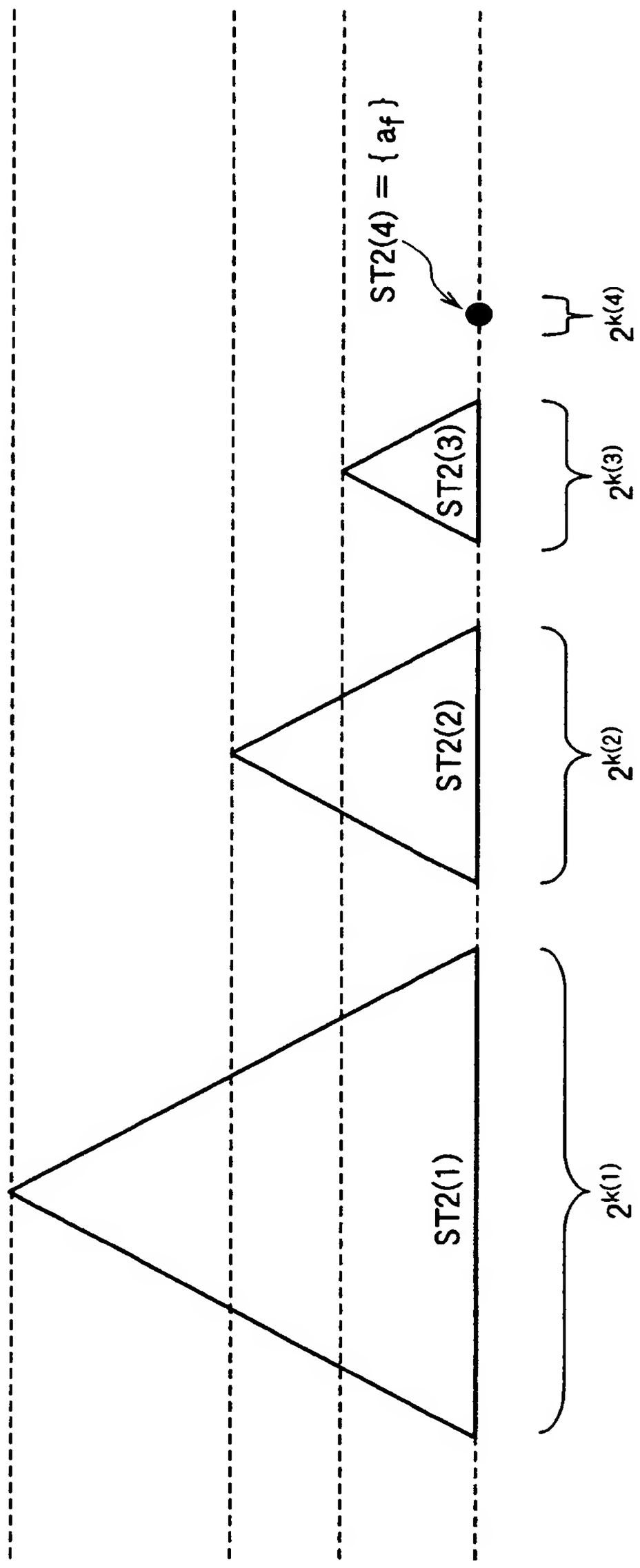

開始
id=0,....., N-1について以下のブロック1を実行する
ST1221a
ブロック1
ブロック開始
ST1222a
(1) 局所変数chain_comple_data2に_chain_comple_data_vec[id]を設定する
(2) 局所変数prev_chain_point2にchain_comple_data2のprev_point部を設定する
ST1223a
prev_chain_point2はnilか?
YES
NO
ST1224a
(1) chain_comple_data2のprev_point_old部をprev_chain_point2に設定
(2) chain_comple_data2のprev_point部をnilに設定
(3) chain_comple_data2のold_tree部に現在の集約木識別子を設定する
(4) _chain_comple_data_vec[id]にchain_comple_data2を設定する
ブロック正常終了
正常終了

[図57]

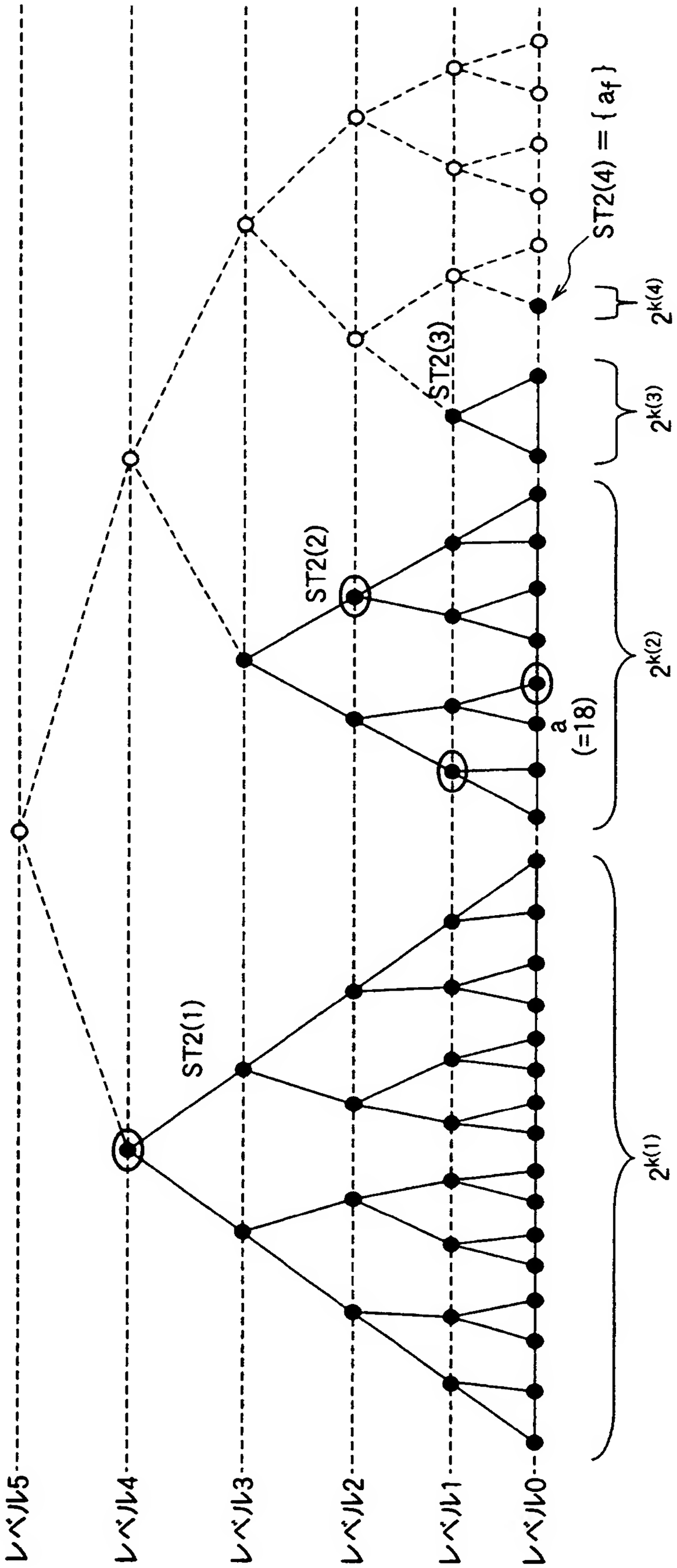

_stack の内容の変化
[(1, 4), V(1, 4)]
[(3, 0), V(3, 0)]
[(3, 0), V(3, 0)]
[(2, 2), V(2, 2)]
[(3, 0), V(3, 0)]
[(3, 0), V(3, 0)]
[(3, 1), V(3, 1)]
[(3, 0), V(3, 0)]
[(4, 0), V(4, 0)]
(1)
(2)
(3)
(4)


終端ポイント
レベル4
レベル3
レベル2
レベル1
レベル0

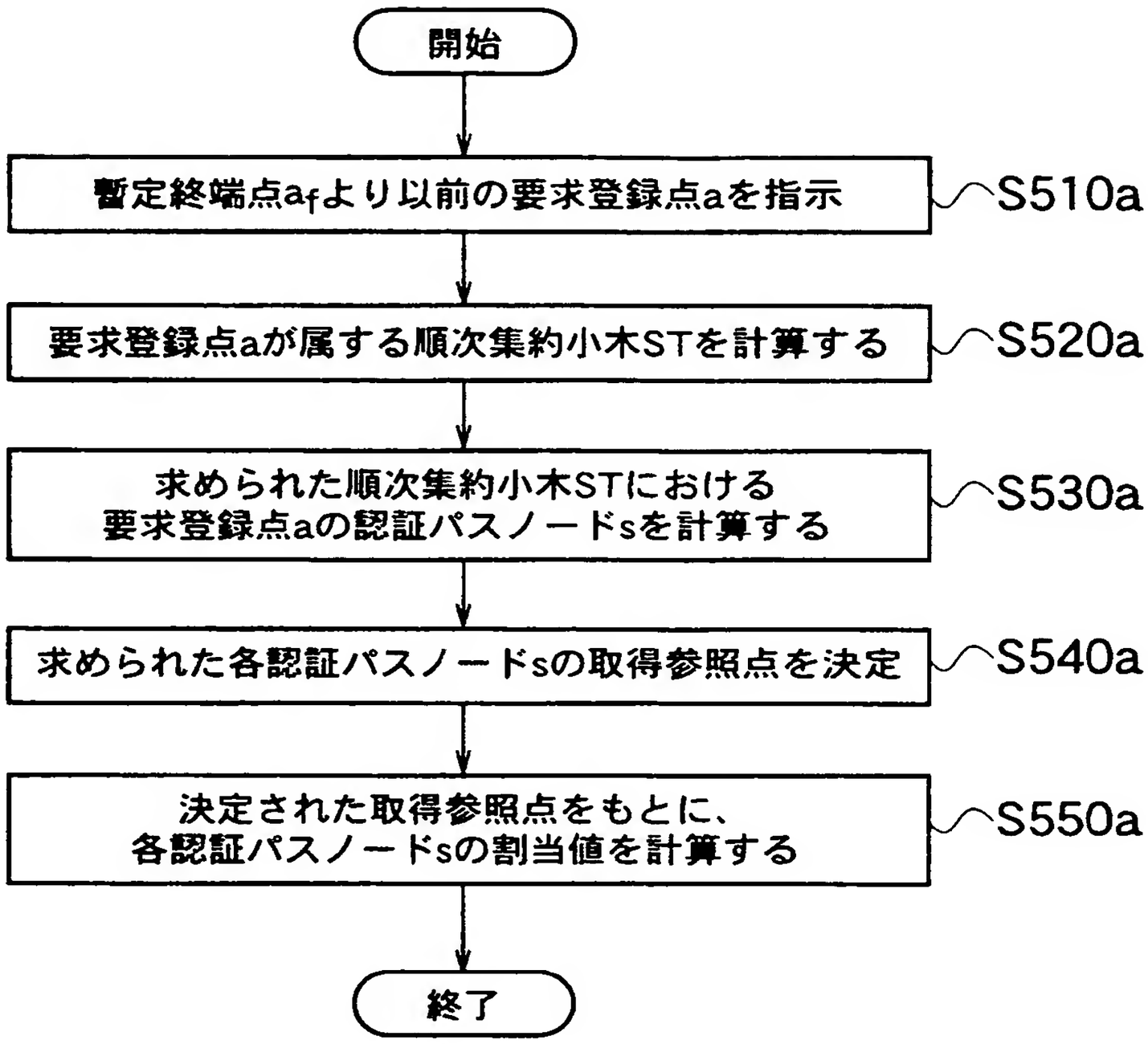
[図58]



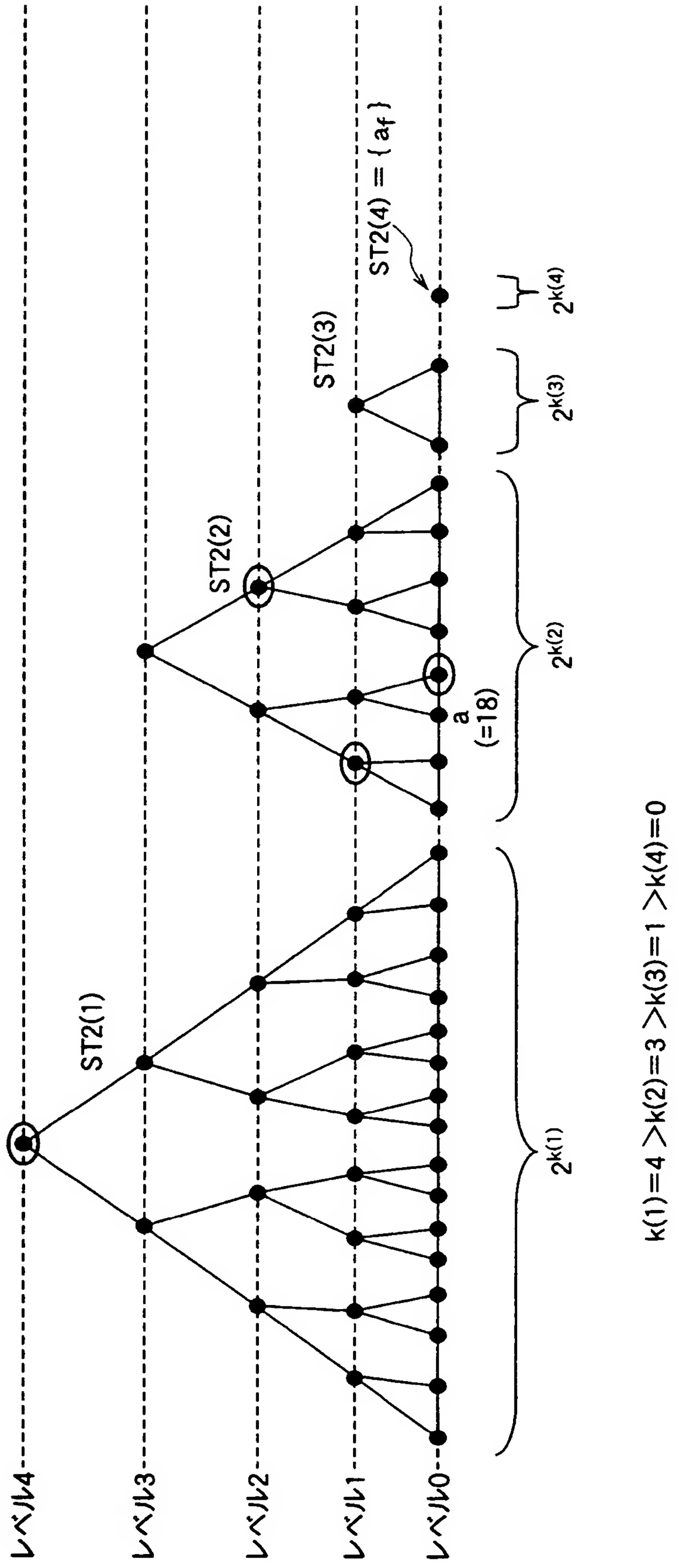
[図59]


レベル5
レベル4
レベル3
レベル2
レベル1
レベル0
ST2(1)
ST2(2)
ST2(3)
ST2(4) = {af}
a
(=18)
2k(1)
2k(2)
2k(3)
2k(4)
k(1)=4 >k(2)=3 >k(3)=1 >k(4) = 0

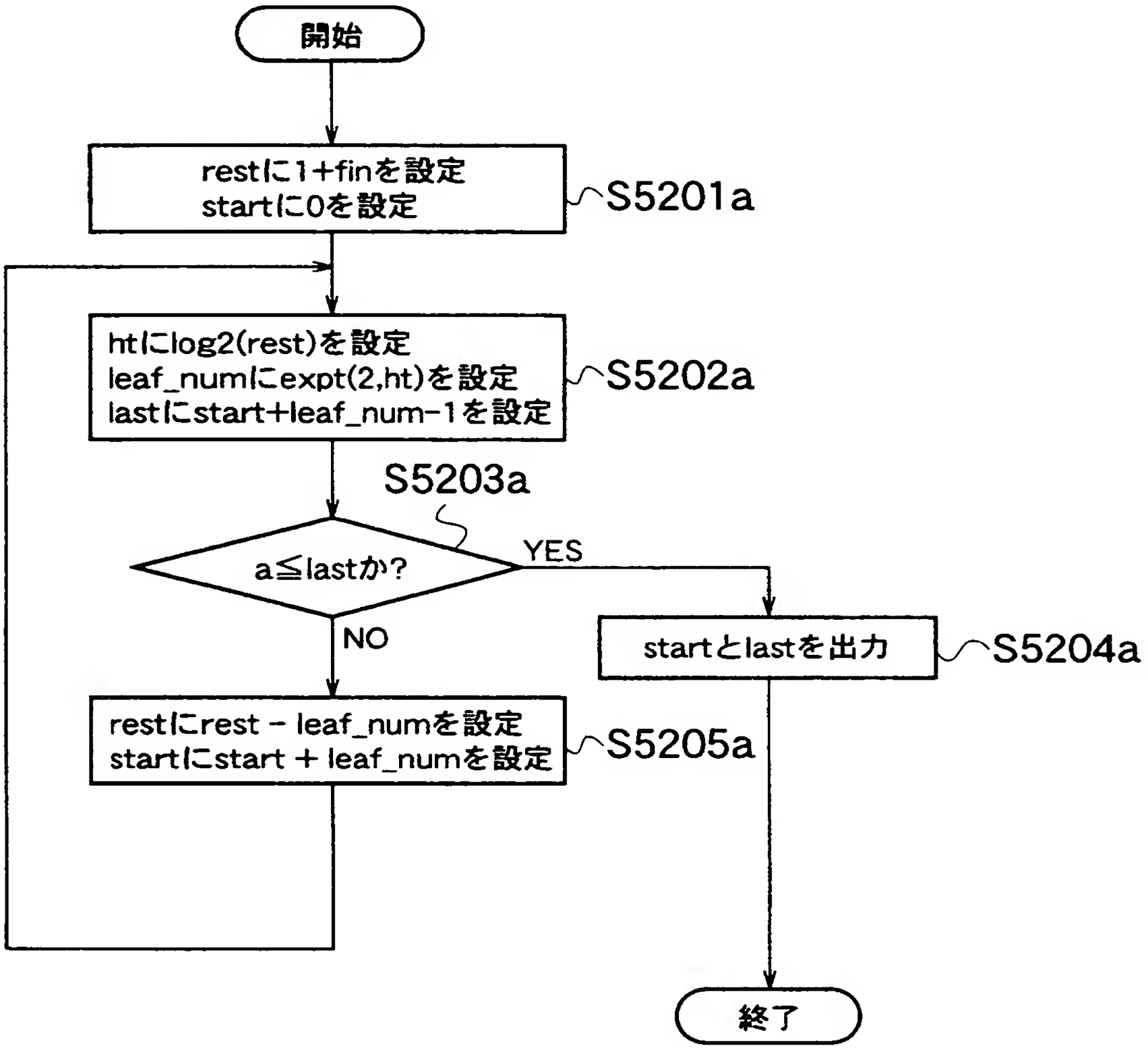
[図60]


開始
暫定終端点afより以前の要求登録点aを指示
S510a
要求登録点aが属する順次集約小木STを計算する
S520a
求められた順次集約小木STにおける
要求登録点aの認証パスノードsを計算する
S530a
求められた各認証パスノードsの取得参照点を決定
S540a
決定された取得参照点をもとに、
各認証パスノードsの割当値を計算する
S550a
終了

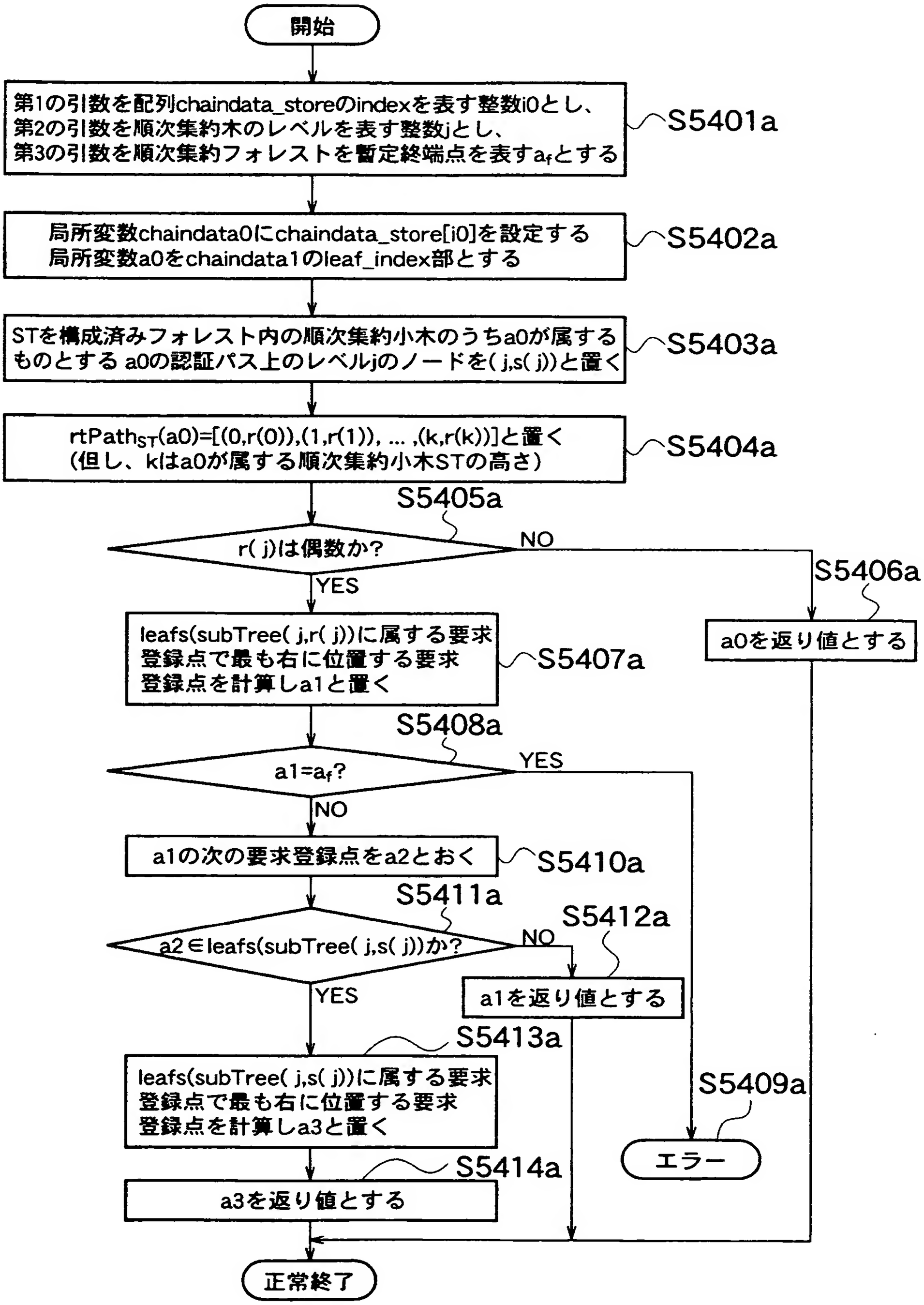
[図61]


ST2(4) = { af }
ST2(3)
ST2(2)
ST2(1)
a
(=18)
2k(4)
2k(3)
2k(2)
2k(1)
レベル4
レベル3
レベル2
レベル1
レベル0
k(1)=4 >k(2)=3 >k(3)=1 >k(4)=0

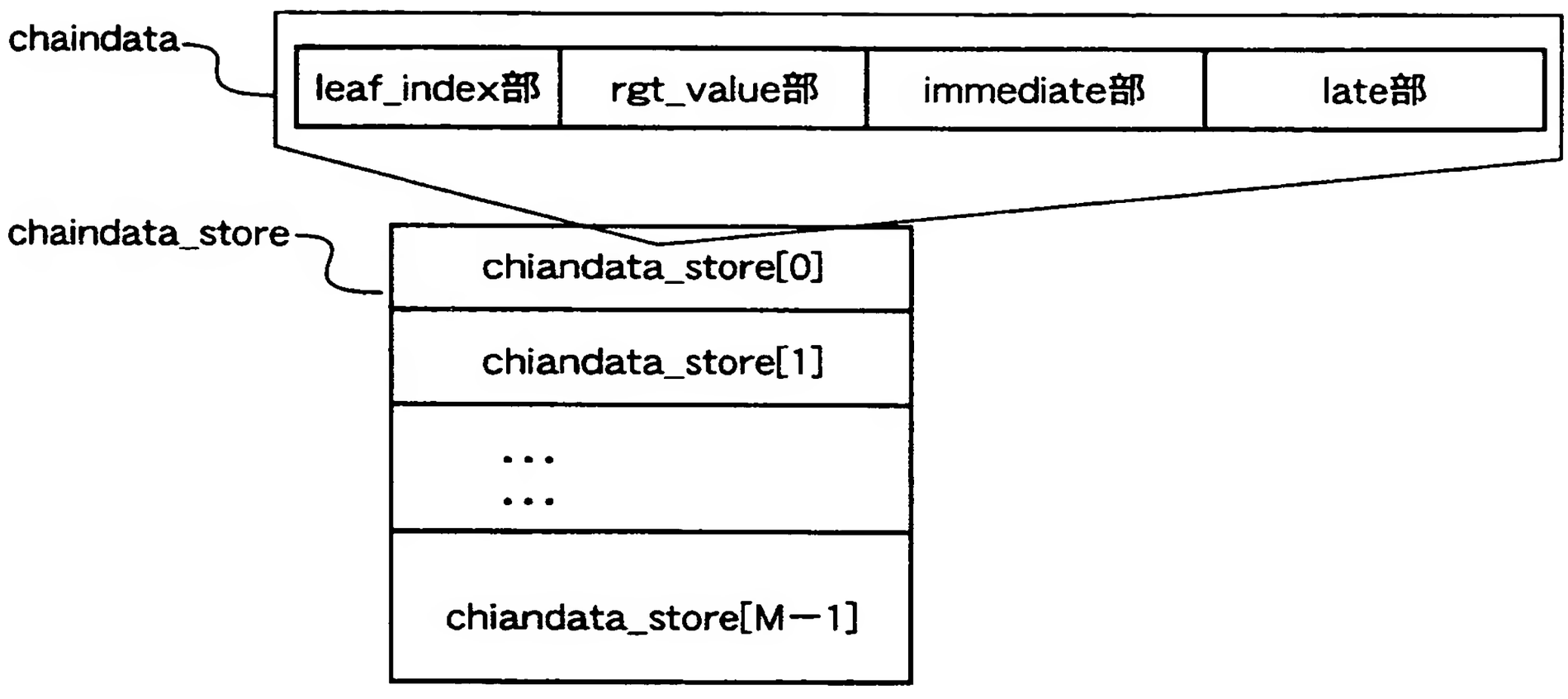
[図62]


開始
restに1+finを設定
startに0を設定
S5201a
htにlog2(rest)を設定
leaf_numにexpt(2,ht)を設定
lastにstart+leaf_num-1を設定
S5202a
S5203a
a≦lastか?
YES
NO
startとlastを出力
S5204a
restにrest - leaf_numを設定
startにstart + leaf_numを設定
S5205a
終了

[図63]


開始
第1の引数を配列chaindata_storeのindexを表す整数i0とし、
第2の引数を順次集約木のレベルを表す整数jとし、
第3の引数を順次集約フォレストを暫定終端点を表すafとする
S5401a
局所変数chaindata0にchaindata_store[i0]を設定する
局所変数a0をchaindata1のleaf_index部とする
S5402a
STを構成済みフォレスト内の順次集約小木のうちa0が属する
ものとする a0の認証パス上のレベルjのノードを(j,s(j))と置く
S5403a
rtPathST(a0)=[(0,r(0)),(1,r(1)), ... ,(k,r(k))]と置く
(但し、kはa0が属する順次集約小木STの高さ)
S5404a
S5405a
r(j)は偶数か?
NO
S5406a
YES
a0を返り値とする
leafs(subTree(j,r(j))に属する要求
登録点で最も右に位置する要求
登録点を計算しa1と置く
S5407a
S5408a
a1=af?
YES
NO
a1の次の要求登録点をa2とおく
S5410a
S5411a
a2∈leafs(subTree(j,s(j))か?
S5412a
NO
YES
a1を返り値とする
S5413a
leafs(subTree(j,s(j))に属する要求
登録点で最も右に位置する要求
登録点を計算しa3と置く
S5409a
S5414a
エラー
a3を返り値とする
正常終了

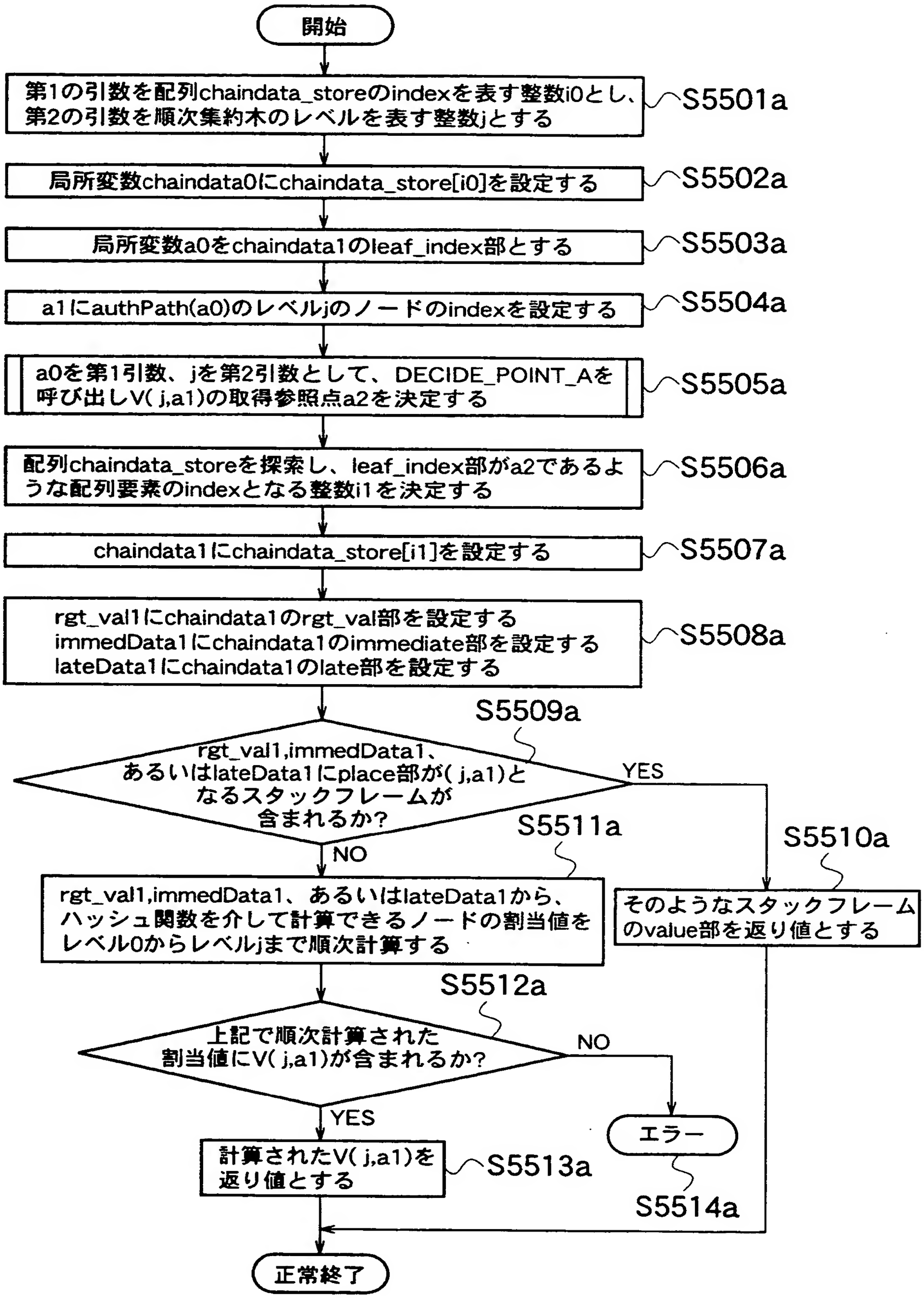
[図64]


chaindata
leaf_index部
rgt_value部
immediate部
late部
chaindata_store
chiandata_store[0]
chiandata_store[1]
・・・
・・・
chiandata_store[M−1]

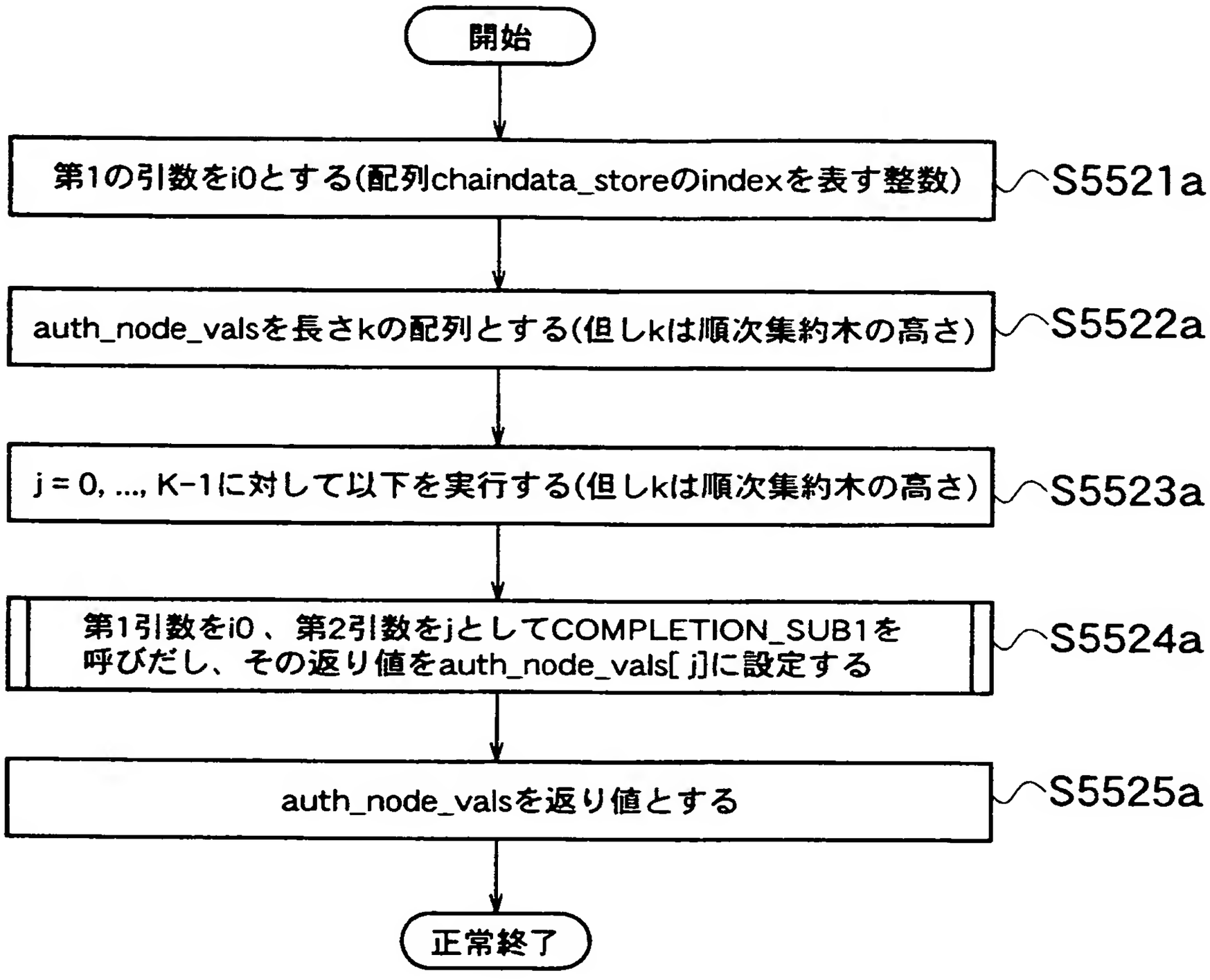
[図65]

(a)
(j+1, r(j+1))
(j, r(j))
(j, s(j))
a0
a1
a2
a3
取得参照点
取得タイミング点
(b)
(j+1, r(j+1))
(j, r(j))
(j, s(j))
a0
a1
a2
a3
a4
取得参照点
取得タイミング点
(追伸点である場合も含む)
(c)
(j+1, r(j+1))
(j, r(j))
(j, s(j))
a0
a1
a2
a3=af
last(subTree(j, s(j)))
(d)
(j+1, r(j+1))
(j, r(j))
(j, s(j))
a0
a1
取得参照点
a2
取得タイミング点
(追伸点である場合も含む)
(e)
(j+1, r(j+1))
(j, r(j))
(j, s(j))
a0
a1=af
取得参照点
(f)
(j+1, r(j+1))
(j, s(j))
(j, r(j))
a0
取得参照点
取得タイミング点

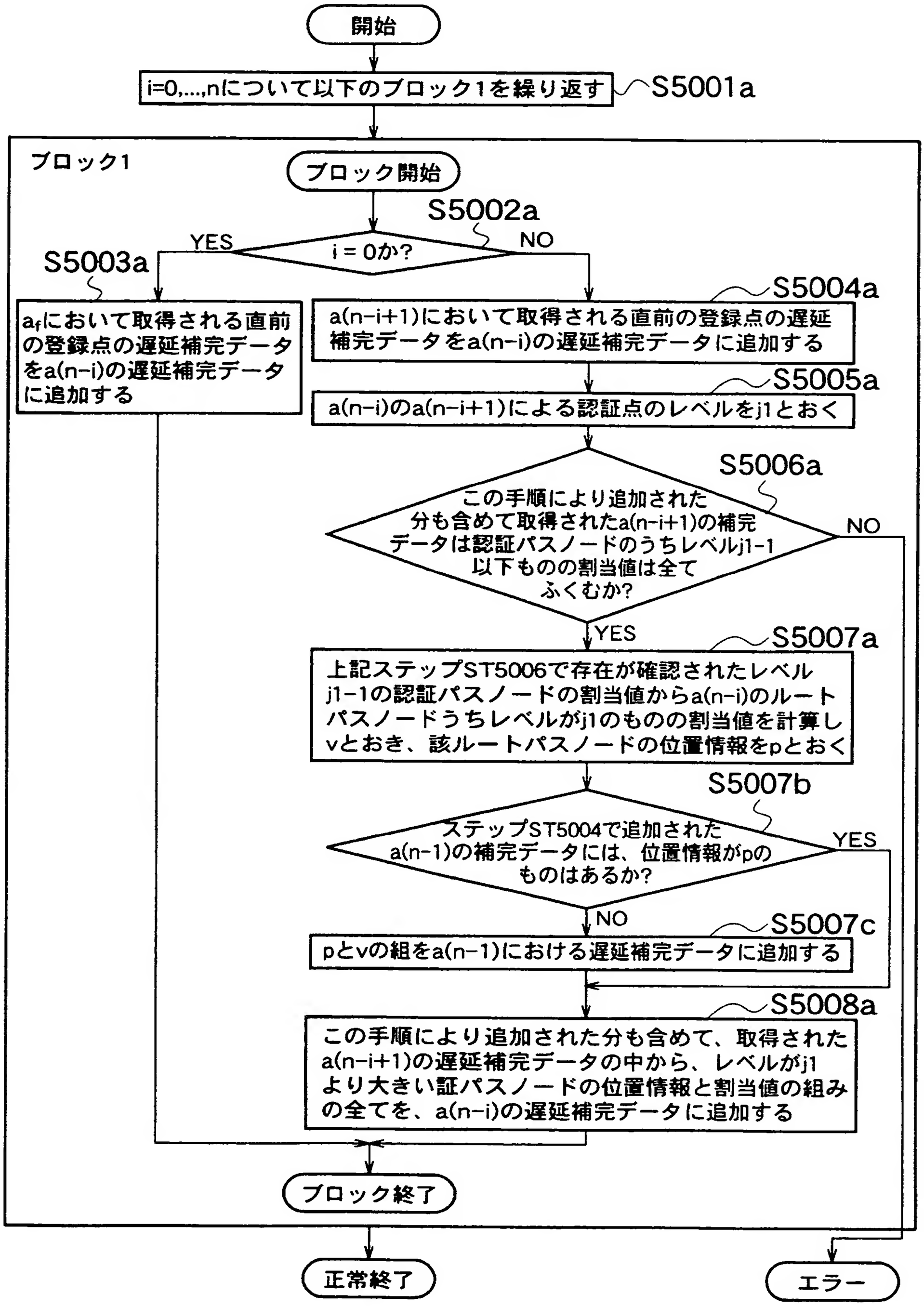
[図66]


開始
第1の引数を配列chaindata_storeのindexを表す整数i0とし、第2の引数を順次集約木のレベルを表す整数jとする
S5501a
局所変数chaindata0にchaindata_store[i0]を設定する
S5502a
局所変数a0をchaindata1のleaf_index部とする
S5503a
a1にauthPath(a0)のレベルjのノードのindexを設定する
S5504a
a0を第1引数、jを第2引数として、DECIDE_POINT_Aを呼び出しV(j,a1)の取得参照点a2を決定する
S5505a
配列chaindata_storeを探索し、leaf_index部がa2であるような配列要素のindexとなる整数i1を決定する
S5506a
chaindata1にchaindata_store[i1]を設定する
S5507a
rgt_val1にchaindata1のrgt_val部を設定する
immedData1にchaindata1のimmediate部を設定する
lateData1にchaindata1のlate部を設定する
S5508a
S5509a
rgt_val1,immedData1、あるいはlateData1にplace部が(j,a1)となるスタックフレームが含まれるか?
YES
NO
S5510a
そのようなスタックフレームのvalue部を返り値とする
S5511a
rgt_val1,immedData1、あるいはlateData1から、ハッシュ関数を介して計算できるノードの割当値をレベル0からレベルjまで順次計算する
S5512a
上記で順次計算された割当値にV(j,a1)が含まれるか?
NO
YES
エラー
S5514a
計算されたV(j,a1)を返り値とする
S5513a
正常終了

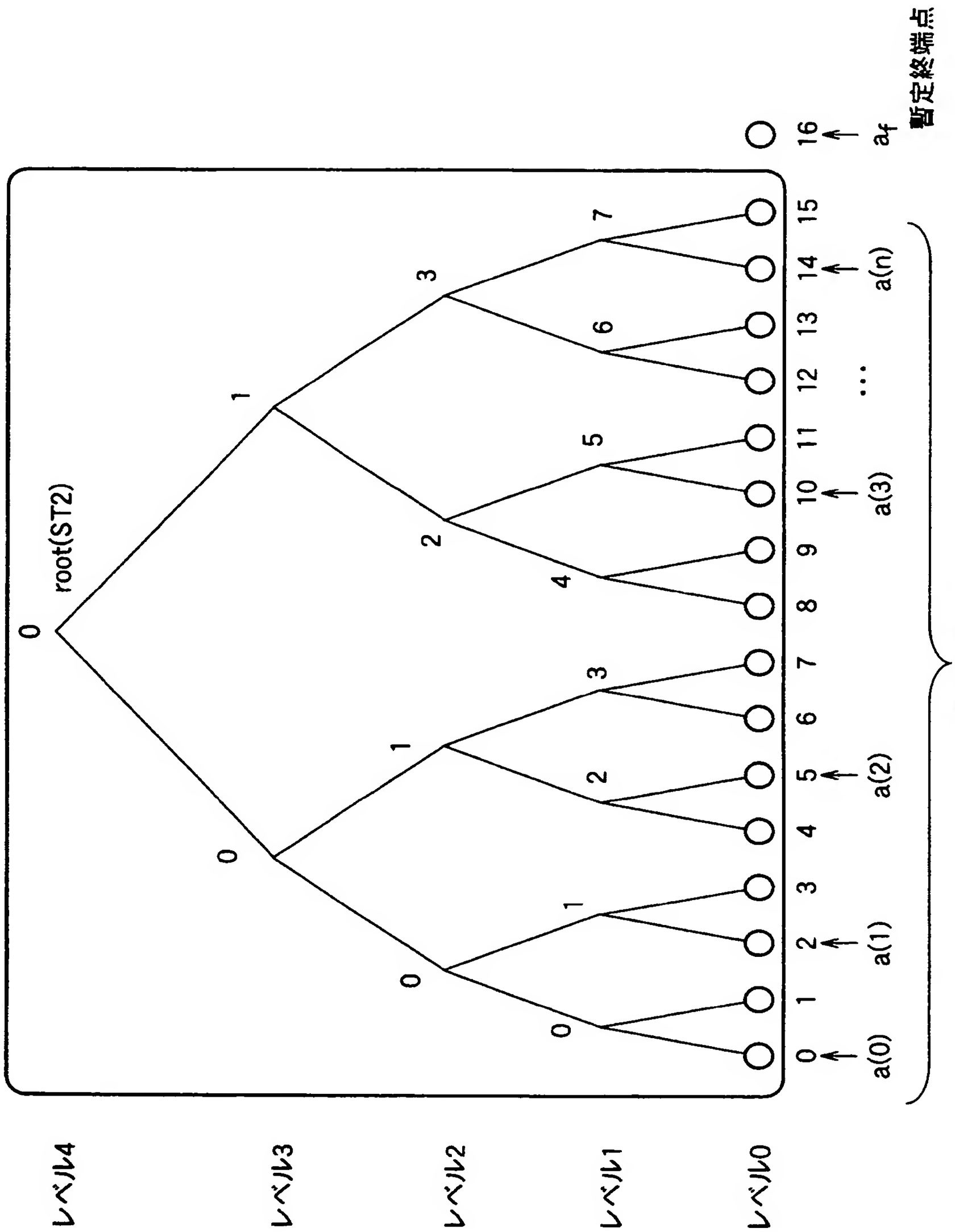
[図67]


開始
第1の引数をi0とする(配列chaindata_storeのindexを表す整数)
S5521a
auth_node_valsを長さkの配列とする(但しkは順次集約木の高さ)
S5522a
j = 0, ..., K-1に対して以下を実行する(但しkは順次集約木の高さ)
S5523a
第1引数をi0 、第2引数をjとしてCOMPLETION_SUB1を
呼びだし、その返り値をauth_node_vals[j]に設定する
S5524a
auth_node_valsを返り値とする
S5525a
正常終了

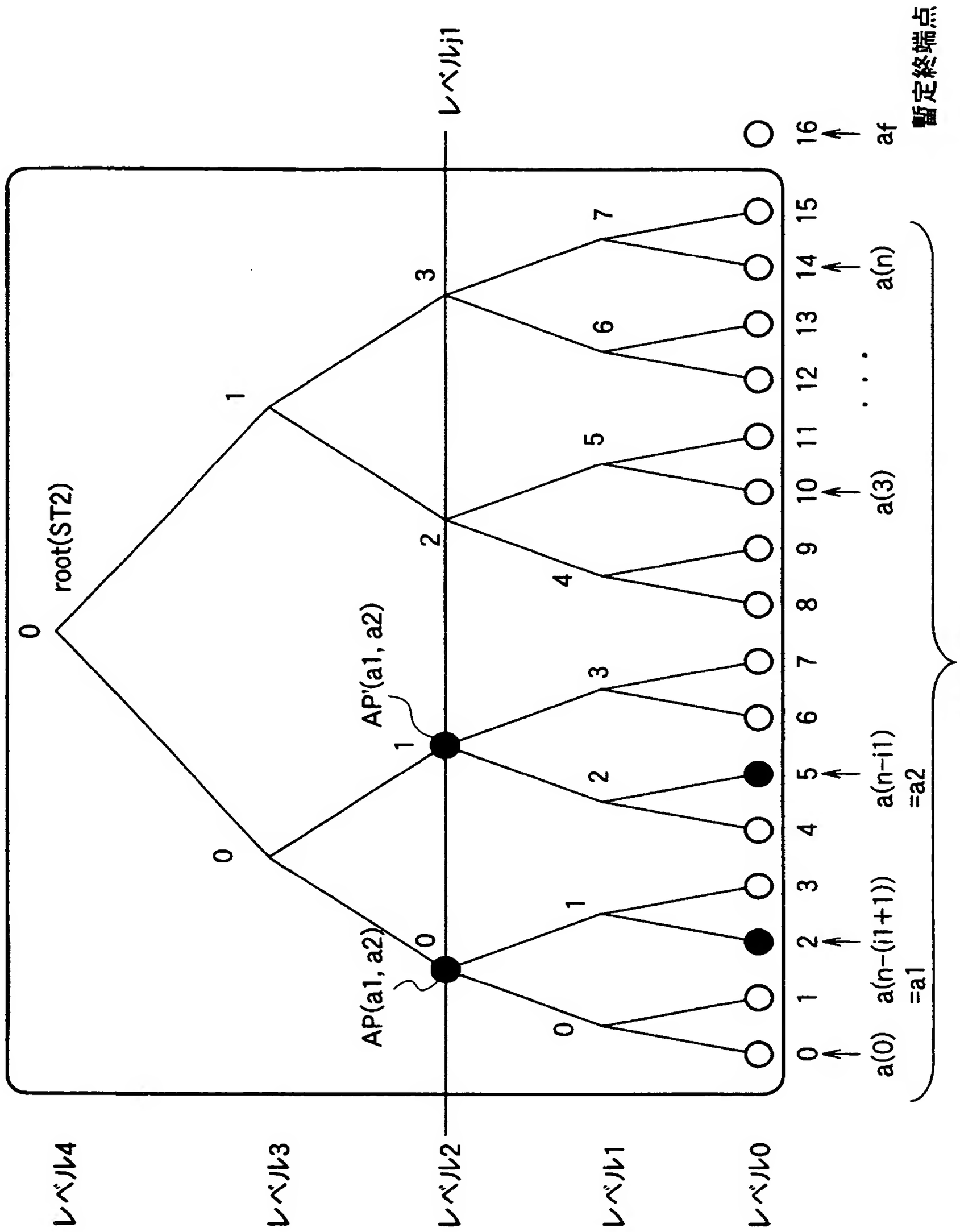
[図68]


開始
i=0,...,nについて以下のブロック1を繰り返す
S5001a
ブロック1
ブロック開始
S5002a
YES
i = 0か?
NO
S5003a
S5004a
a_fにおいて取得される直前の登録点の遅延補完データをa(n-i)の遅延補完データに追加する
a(n-i+1)において取得される直前の登録点の遅延補完データをa(n-i)の遅延補完データに追加する
S5005a
a(n-i)のa(n-i+1)による認証点のレベルをj1とおく
S5006a
この手順により追加された分も含めて取得されたa(n-i+1)の補完データは認証パスノードのうちレベルj1-1以下ものの割当値は全てふくむか?
NO
YES
S5007a
上記ステップST5006で存在が確認されたレベルj1-1の認証パスノードの割当値からa(n-i)のルートパスノードうちレベルがj1のものの割当値を計算しvとおき、該ルートパスノードの位置情報をpとおく
S5007b
ステップST5004で追加されたa(n-1)の補完データには、位置情報がpのものはあるか?
YES
NO
S5007c
pとvの組をa(n-1)における遅延補完データに追加する
S5008a
この手順により追加された分も含めて、取得されたa(n-i+1)の遅延補完データの中から、レベルがj1より大きい証パスノードの位置情報と割当値の組みの全てを、a(n-i)の遅延補完データに追加する
ブロック終了
正常終了
エラー

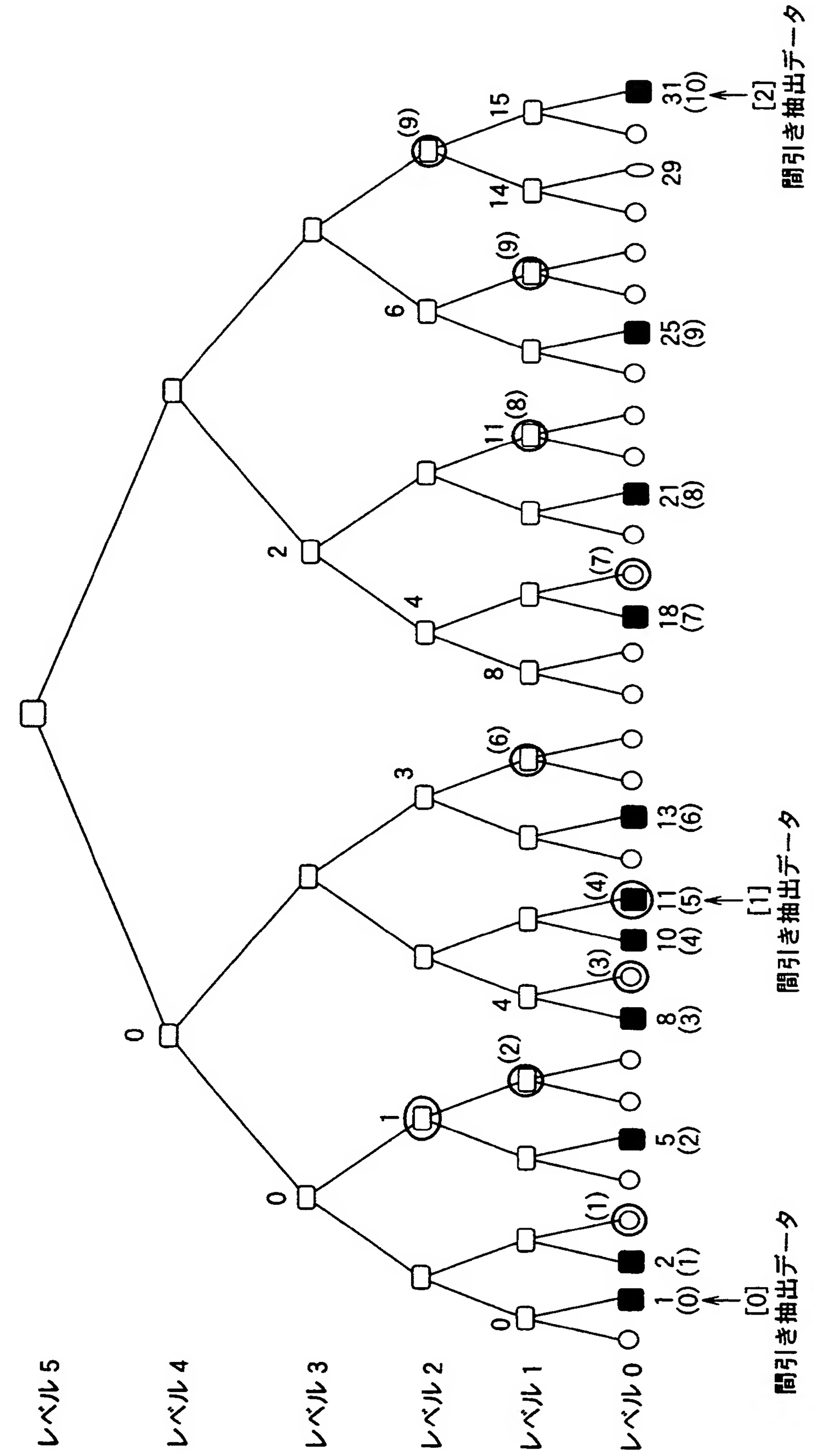
[図69]


順次集約小木ST2
root(ST2)
暫定終端点
af
a(n)
a(3)
a(2)
a(1)
a(0)
登録点集合
レベル4
レベル3
レベル2
レベル1
レベル0

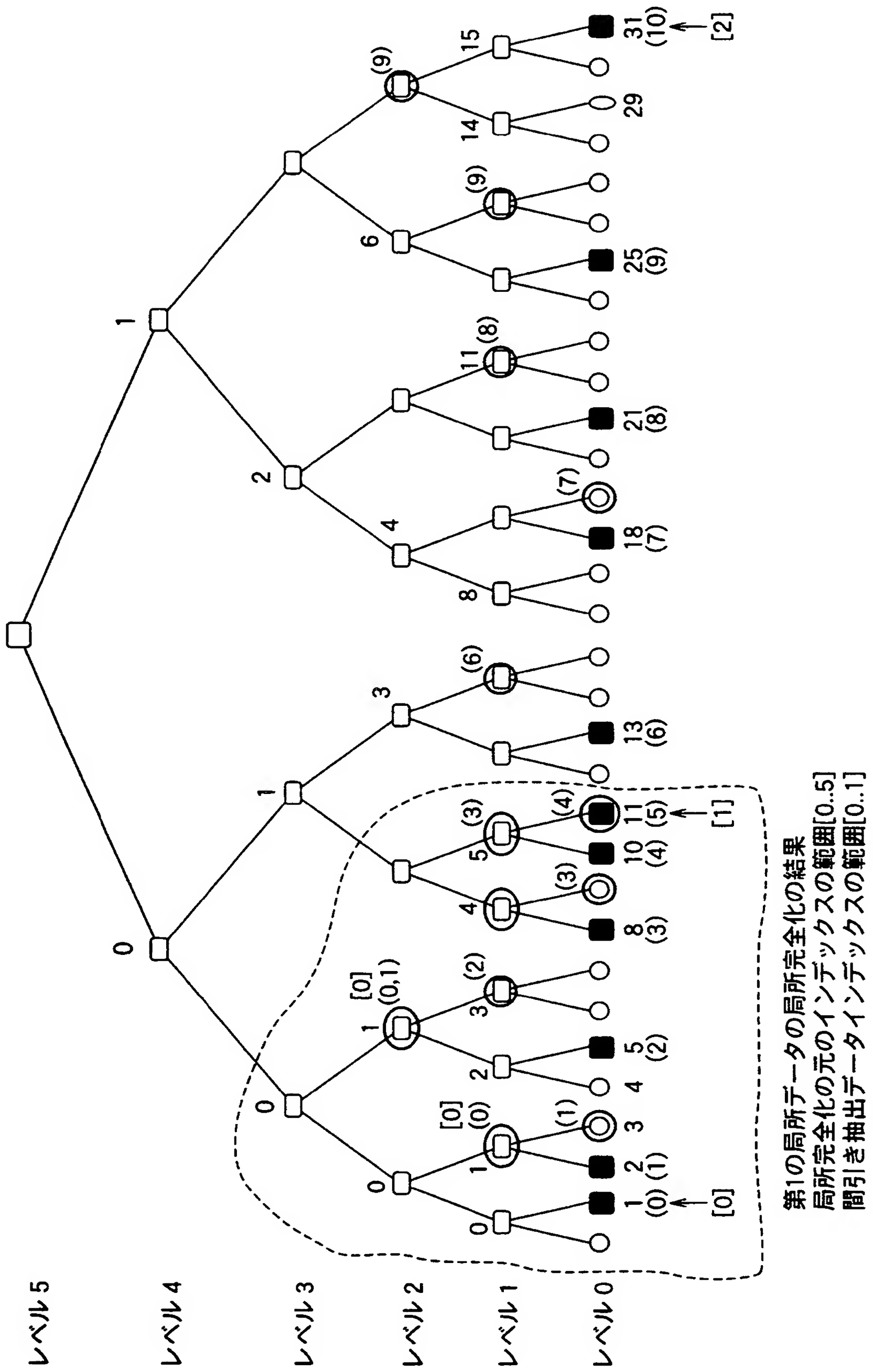
[図70]


順次集約小木ST2
root(ST2)
AP'(a1, a2)
AP(a1, a2)
レベル4
レベル3
レベル2
レベル1
レベル0
a(n)
a(3)
a(n−i1)
=a2
a(n−(i1+1))
=a1
a(0)
af
暫定終端点
登録点集合

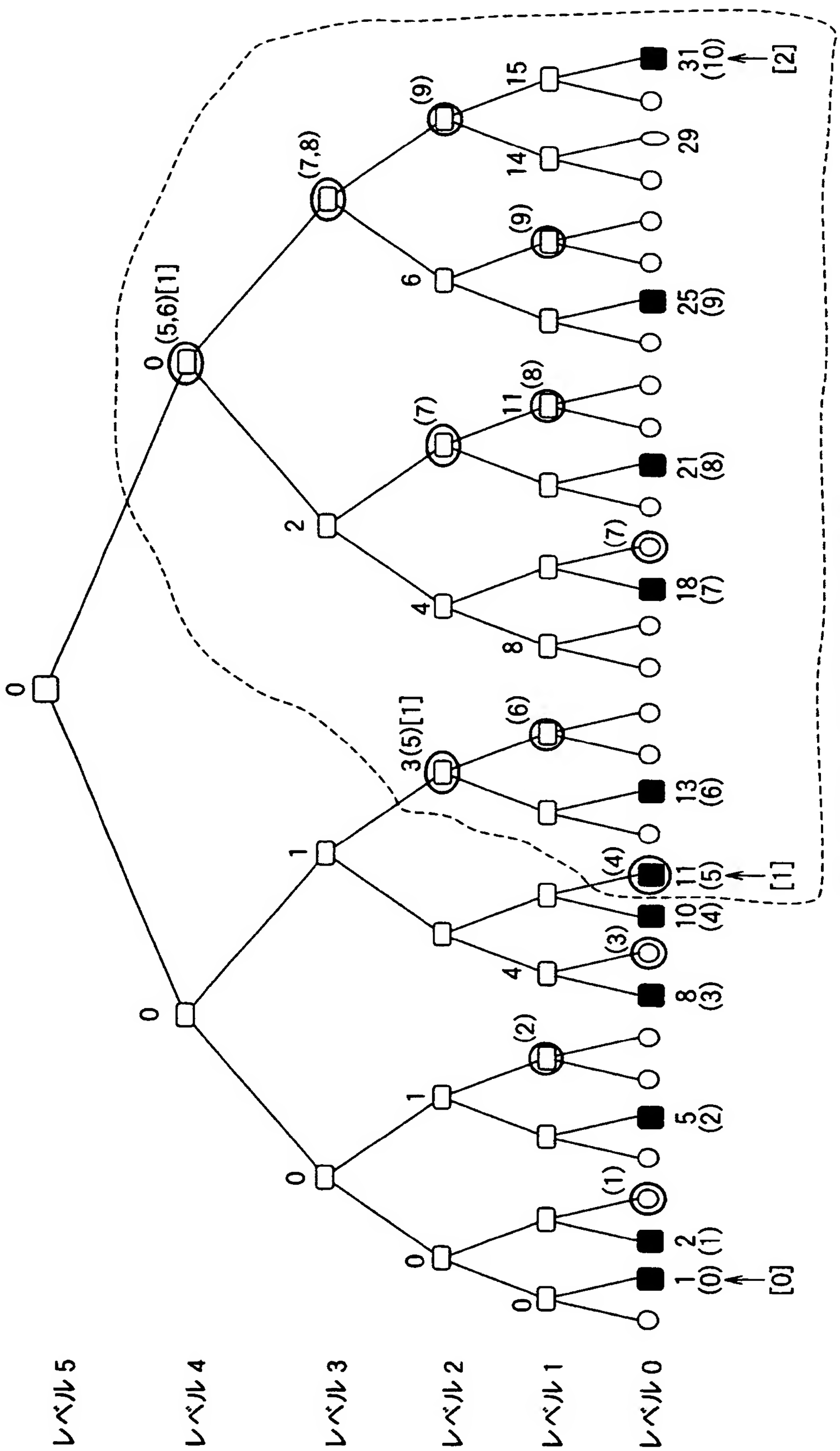
[図71]


レベル5
レベル4
レベル3
レベル2
レベル1
レベル0
0
0
0
1
0
2
0
3
4
0
1
2
3
4
5
6
7
8
(9)
(9)
(9)
(8)
(8)
(7)
(7)
(6)
(6)
(5)
(4)
(4)
(3)
(3)
(2)
(2)
(1)
(1)
(0)
15
14
6
11
21
18
8
13
11
10
4
8
5
2
1
31
(10)
29
25
(9)
21
(8)
18
(7)
13
(6)
11
(5)
10
(4)
8
(3)
5
(2)
2
(1)
1
(0)
[2]
間引き抽出データ
[1]
間引き抽出データ
[0]
間引き抽出データ
間引き抽出データインデックス[0..2]の設定

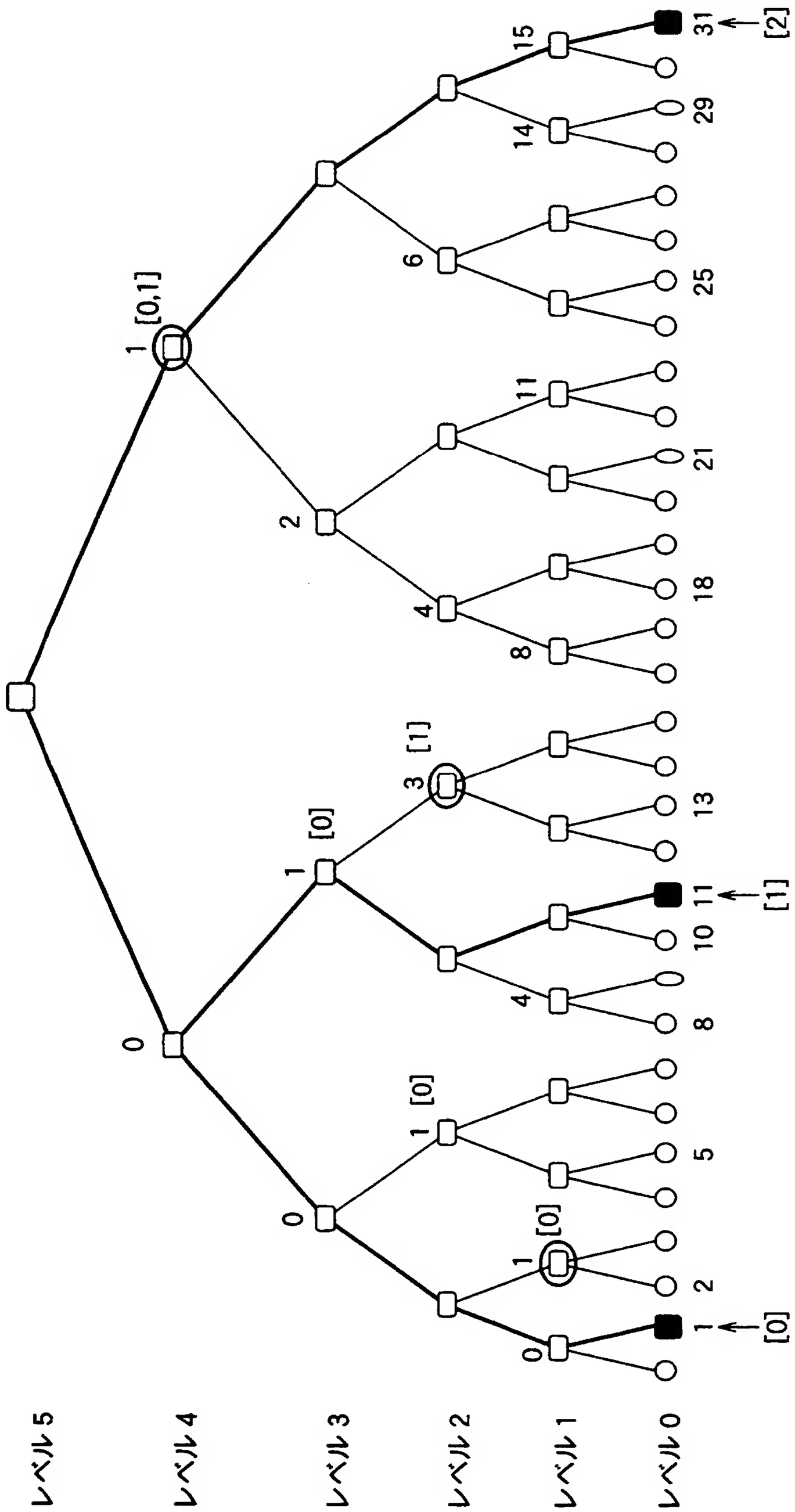
[図72]


レベル5
レベル4
レベル3
レベル2
レベル1
レベル0
[0]
[1]
[2]
第1の局所データの局所完全化の結果
局所完全化の元のインデックスの範囲[0..5]
間引き抽出データインデックスの範囲[0..1]

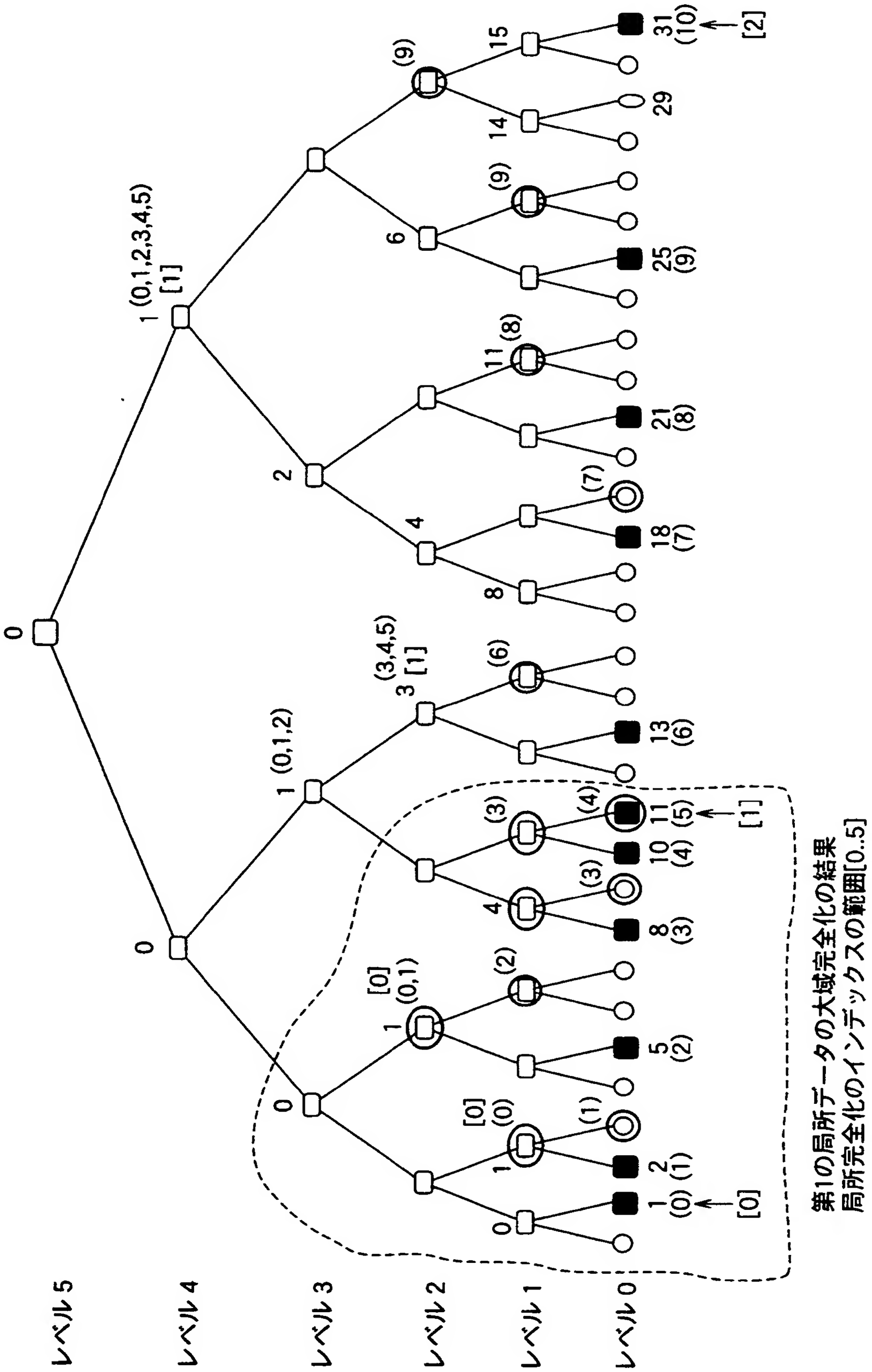
[図73]


レベル5
レベル4
レベル3
レベル2
レベル1
レベル0
第2の局所データの局所完全化の結果
局所完全化のインデックスの範囲[5..10]
間引き抽出データインデックスの範囲[1..2]

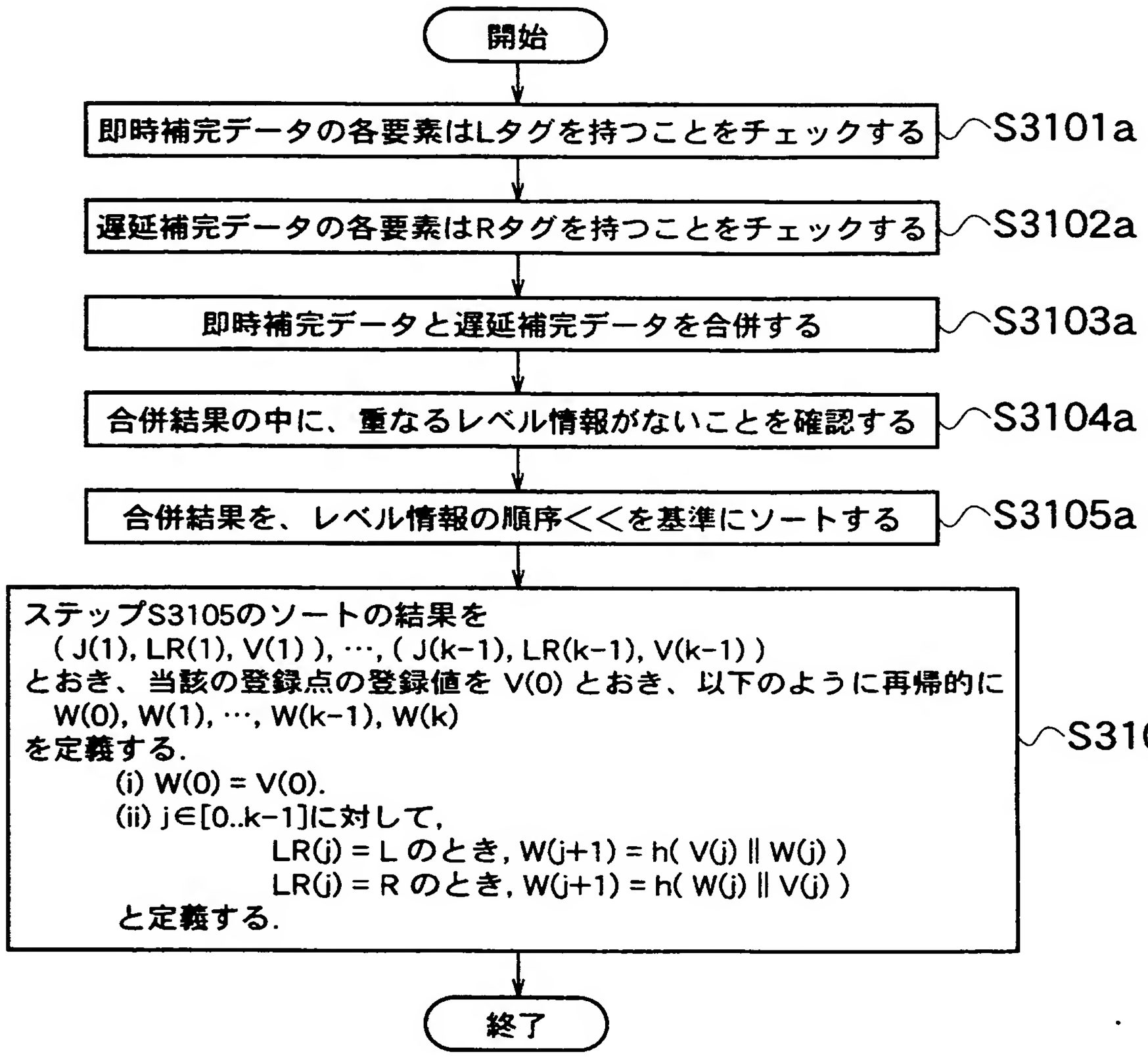
[図74]


レベル5
レベル4
レベル3
レベル2
レベル1
レベル0
0
1 [0,1]
0
2
1 [0]
3 [1]
0
4
1 [0]
2
4
8
1
5
10 11 [1]
8
15
14
6
11
4
8
1 [0]
0
31 ← [2]
29
25
21
18
13
5
2
1 ← [0]

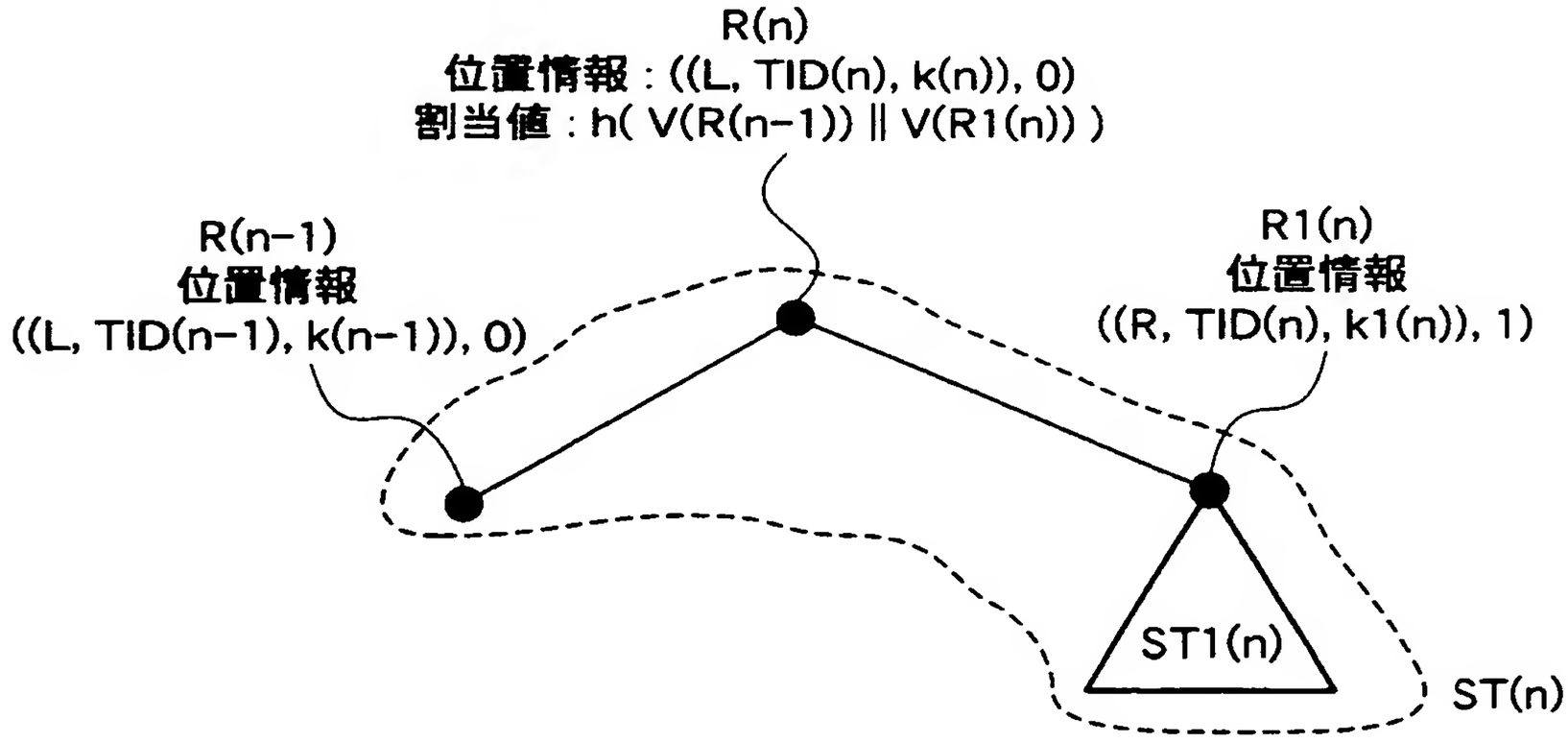
[図75]


レベル5
レベル4
レベル3
レベル2
レベル1
レベル0
第1の局所データの大域完全化の結果
局所完全化のインデックスの範囲[0..5]

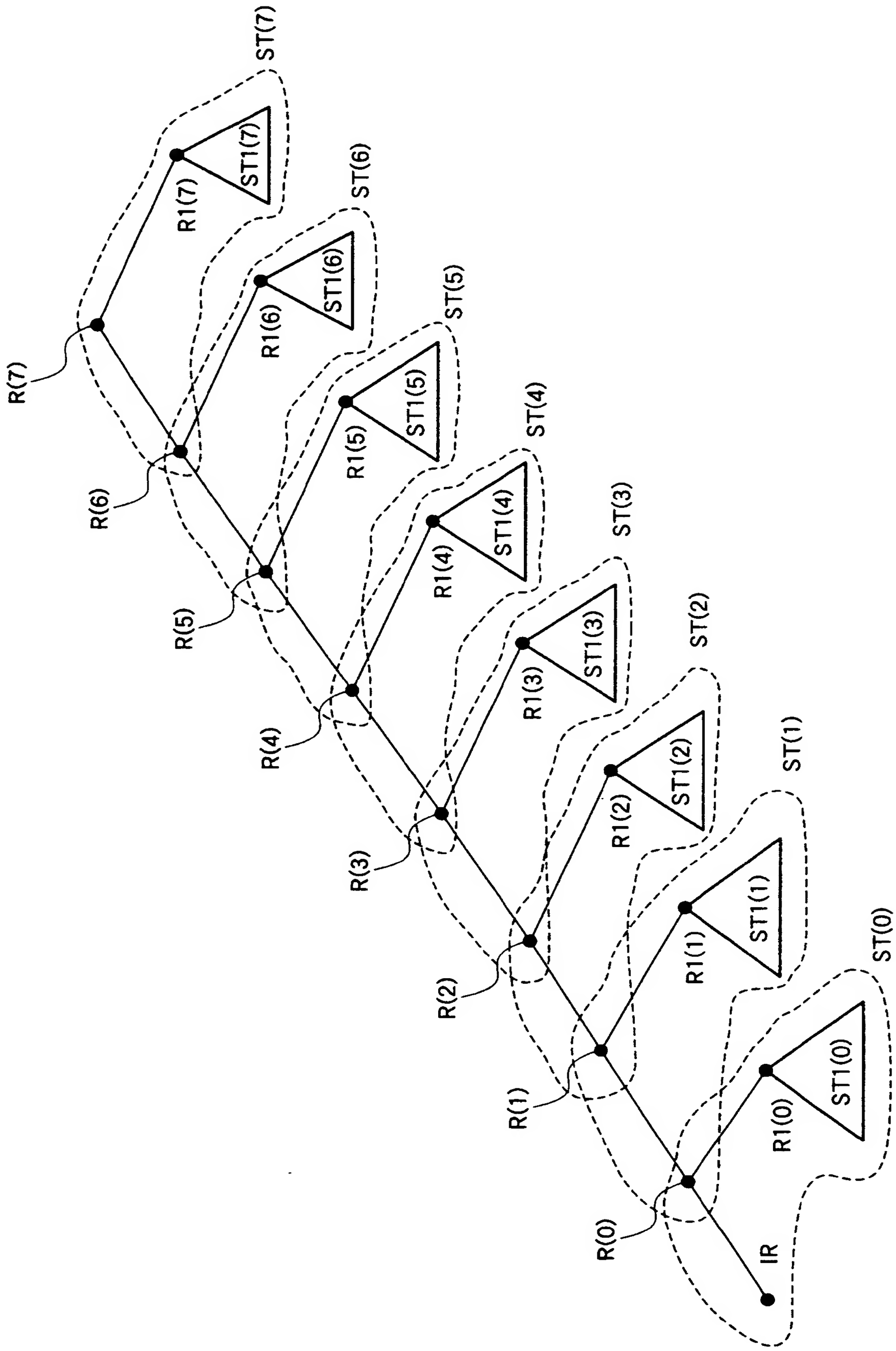
[図76]


開始
即時補完データの各要素はLタグを持つことをチェックする
S3101a
遅延補完データの各要素はRタグを持つことをチェックする
S3102a
即時補完データと遅延補完データを合併する
S3103a
合併結果の中に、重なるレベル情報がないことを確認する
S3104a
合併結果を、レベル情報の順序<<を基準にソートする
S3105a
ステップS3105のソートの結果を
(J(1), LR(1), V(1)), …, (J(k-1), LR(k-1), V(k-1))
とおき、当該の登録点の登録値を V(0) とおき、以下のように再帰的に
W(0), W(1), …, W(k-1), W(k)
を定義する.
(i) W(0) = V(0).
(ii) j∈[0..k-1]に対して,
LR(j) = L のとき, W(j+1) = h(V(j) || W(j))
LR(j) = R のとき, W(j+1) = h(W(j) || V(j))
と定義する.
S3106a
終了

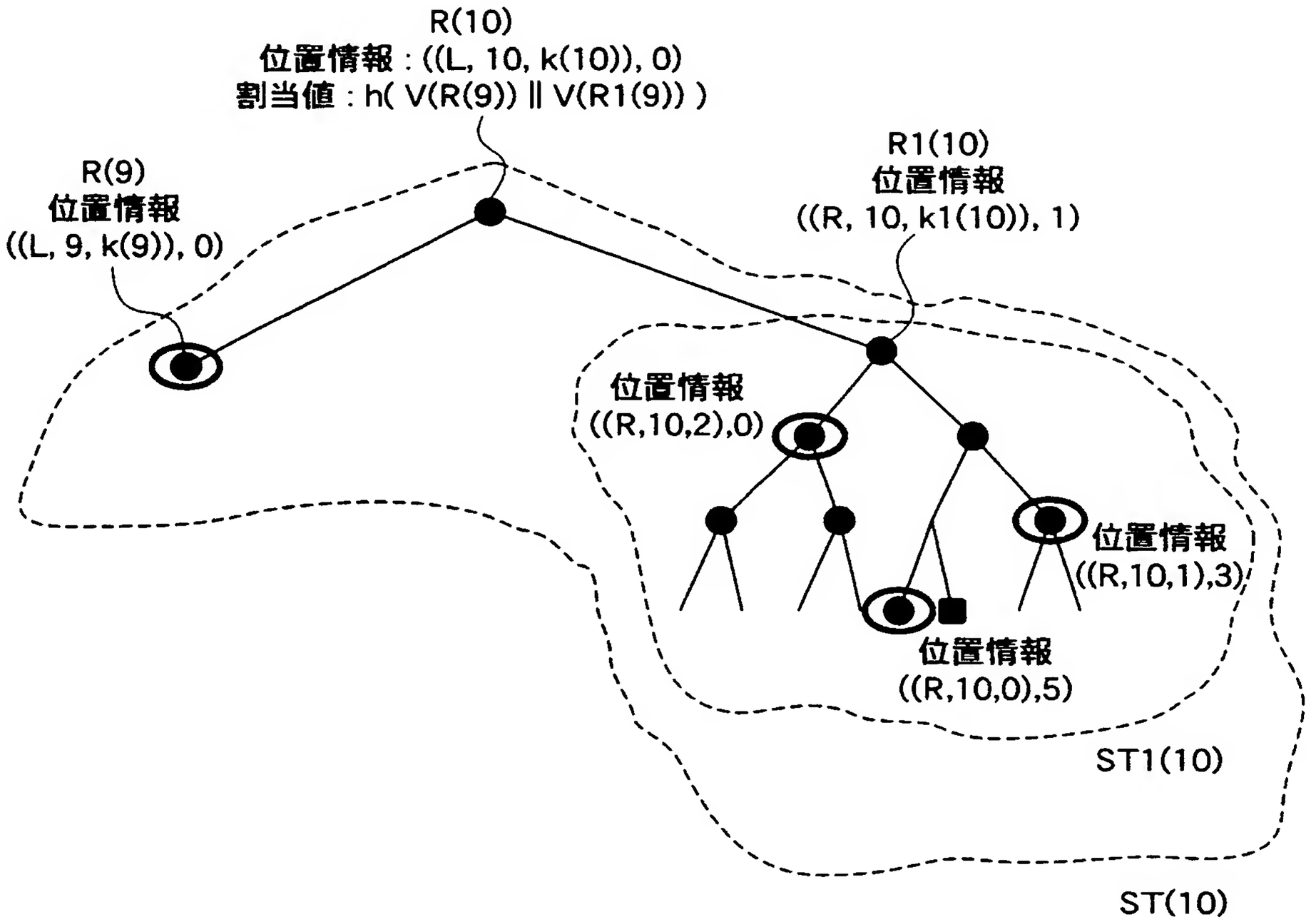
[図77]


R(n)
位置情報 : ((L, TID(n), k(n)), 0)
割当値 : h(V(R(n-1)) || V(R1(n)))
R(n-1)
位置情報
((L, TID(n-1), k(n-1)), 0)
R1(n)
位置情報
((R, TID(n), k1(n)), 1)
ST1(n)
ST(n)

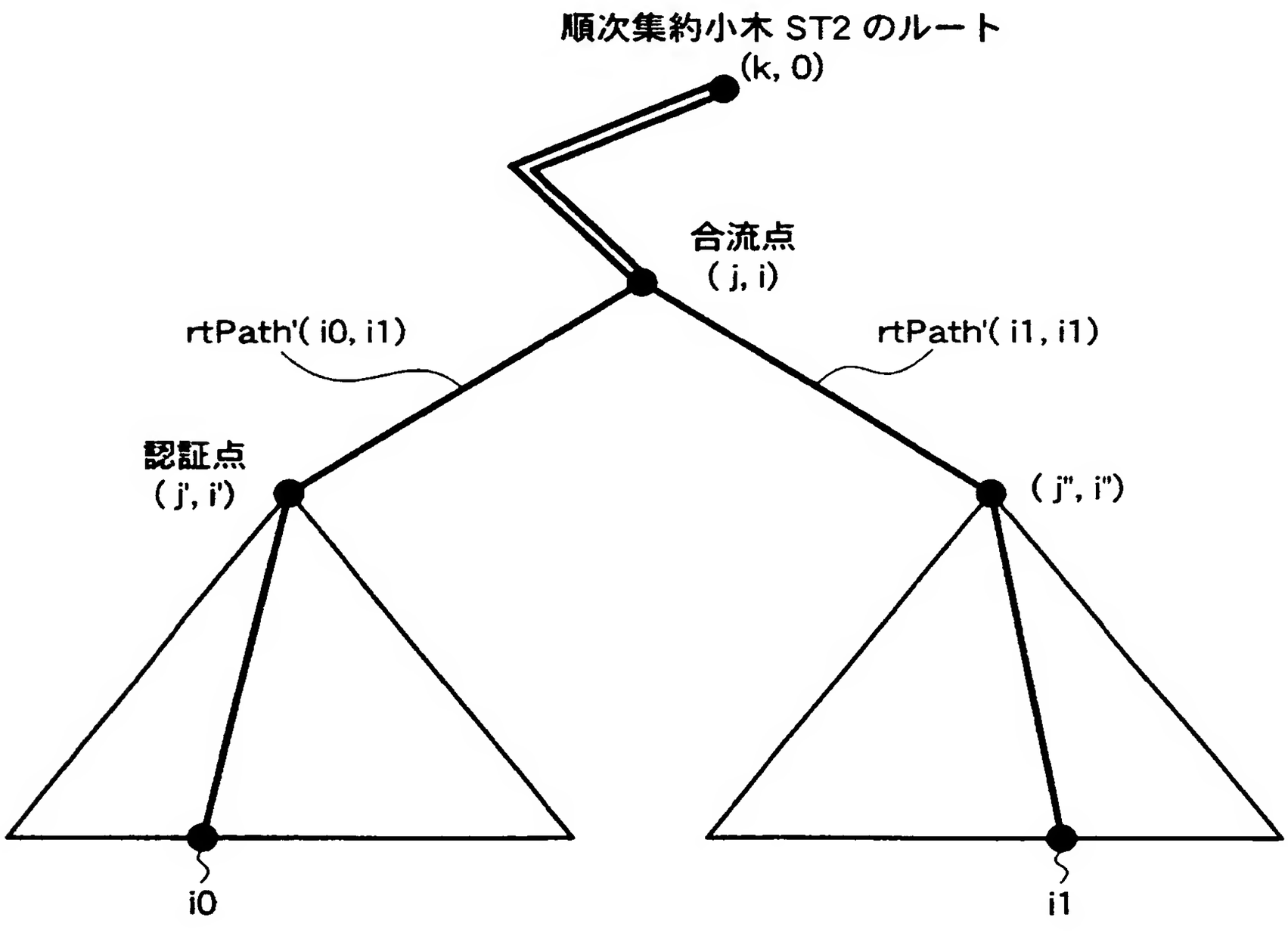
[図78]


ST(7)
ST1(7)
R1(7)
R(7)
ST(6)
ST1(6)
R1(6)
R(6)
ST(5)
ST1(5)
R1(5)
R(5)
ST(4)
ST1(4)
R1(4)
R(4)
ST(3)
ST1(3)
R1(3)
R(3)
ST(2)
ST1(2)
R1(2)
R(2)
ST(1)
ST1(1)
R1(1)
R(1)
ST(0)
ST1(0)
R1(0)
R(0)
IR

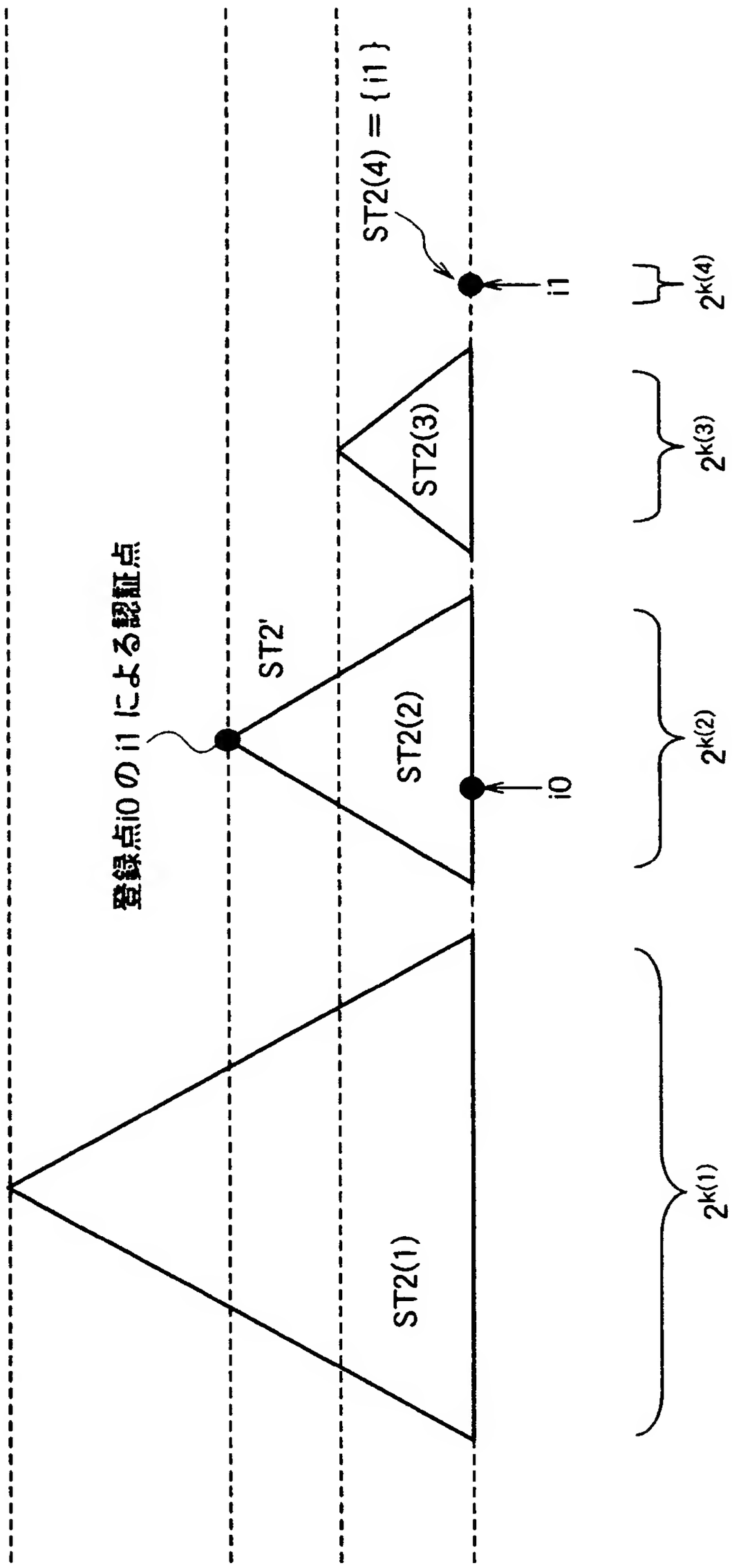
[図79]


R(10)
位置情報：((L, 10, k(10)), 0)
割当値：h(V(R(9)) || V(R1(9)))
R(9)
位置情報
((L, 9, k(9)), 0)
R1(10)
位置情報
((R, 10, k1(10)), 1)
位置情報
((R,10,2),0)
位置情報
((R,10,1),3)
位置情報
((R,10,0),5)
ST1(10)
ST(10)

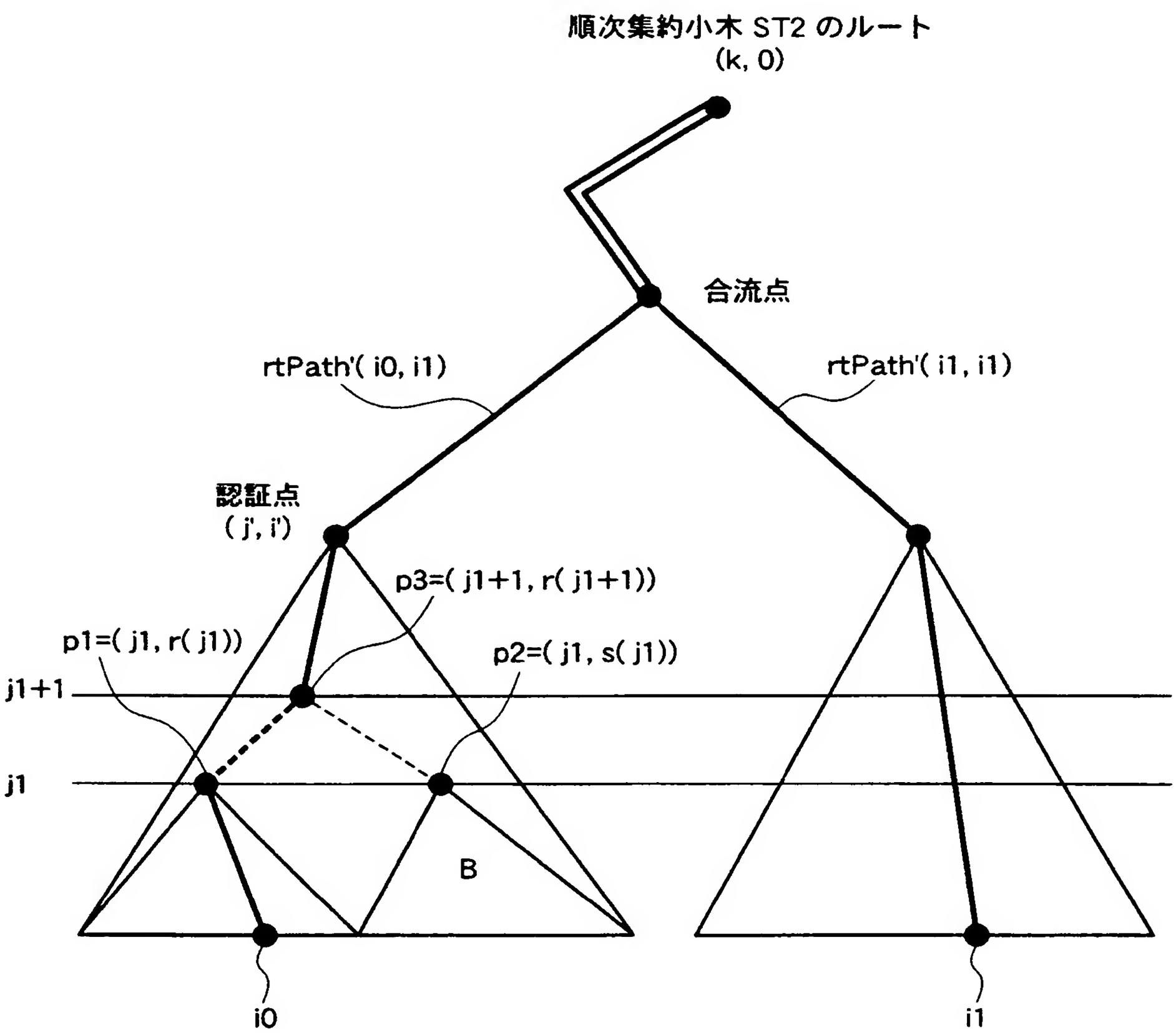
[図80]


順次集約小木 ST2 のルート
(k, 0)
合流点
(j, i)
rtPath'(i0, i1)
rtPath'(i1, i1)
認証点
(j', i')
(j", i")
i0
i1

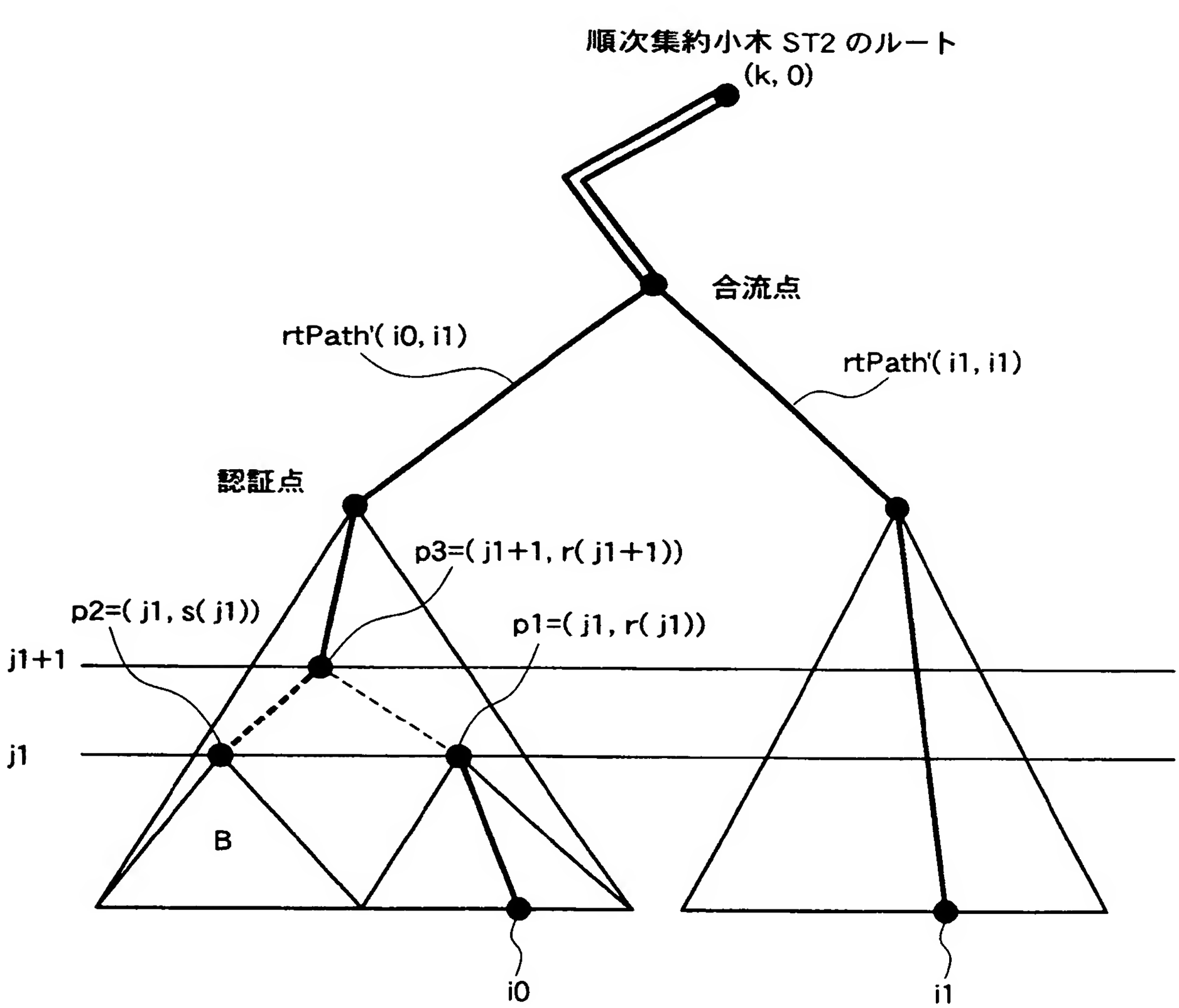
[図81]


ST2(4) = {i1}
i1
$2^{k(4)}$
登録点i0のi1による認証点
ST2'
ST2(3)
ST2(2)
i0
$2^{k(3)}$
$2^{k(2)}$
ST2(1)
$2^{k(1)}$
k(1) >k(2) >k(3) >k(4) = 0

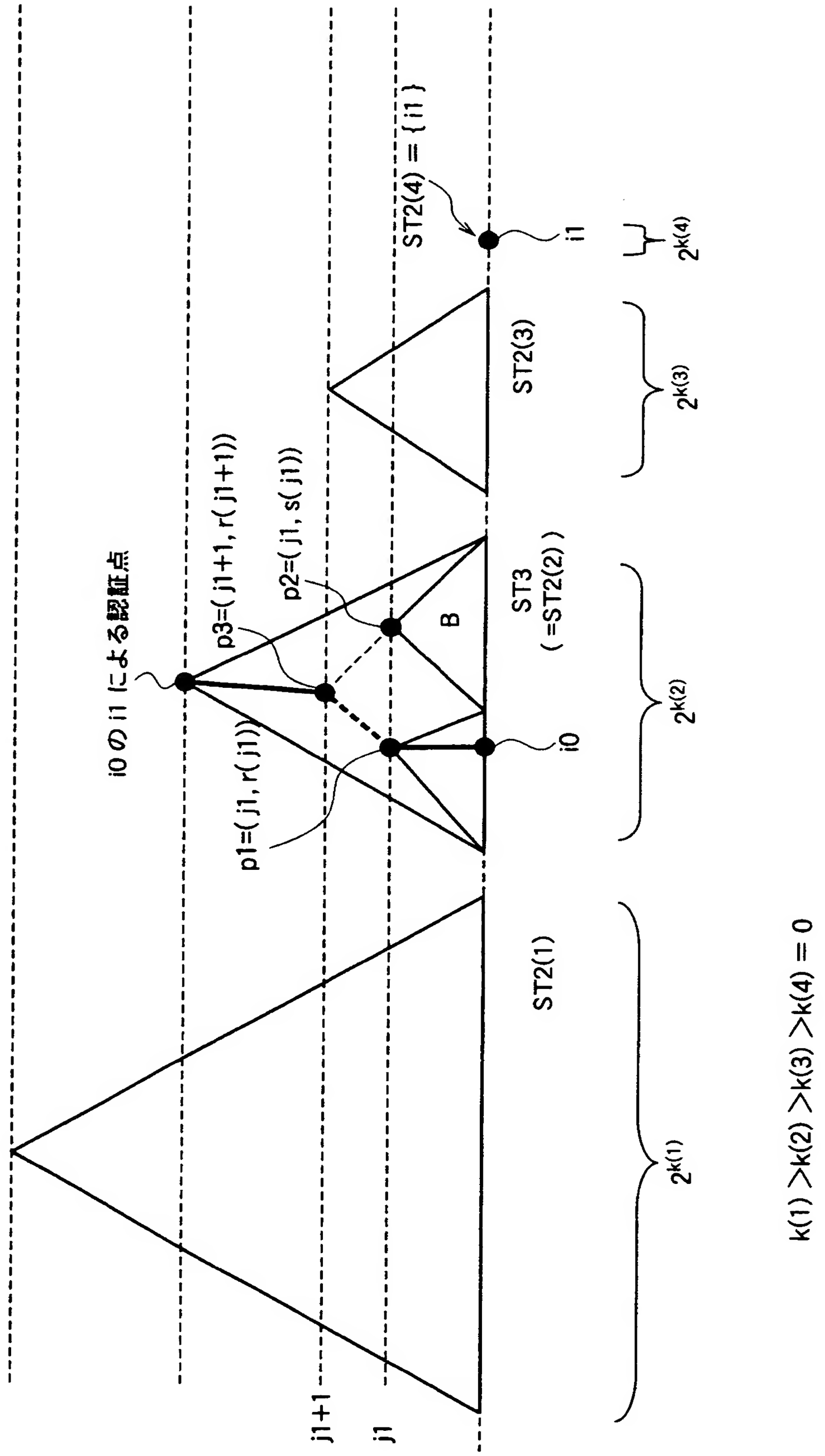
[図82]


順次集約小木 ST2 のルート
(k, 0)
合流点
rtPath'(i0, i1)
rtPath'(i1, i1)
認証点
(j', i')
p3=(j1+1, r(j1+1))
p1=(j1, r(j1))
p2=(j1, s(j1))
j1+1
j1
B
i0
i1

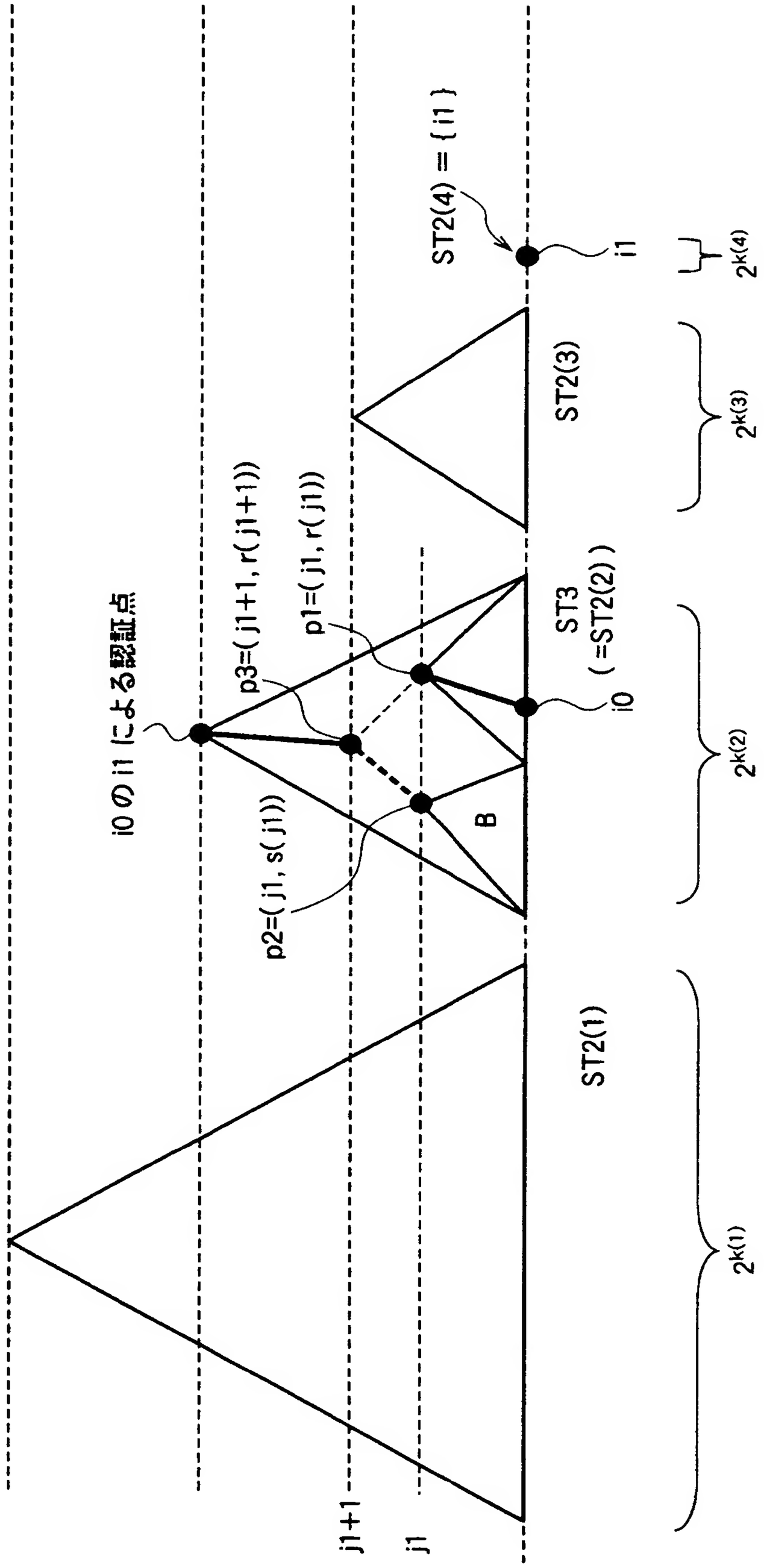
[図83]


順次集約小木 ST2 のルート
(k, 0)
合流点
rtPath'(i0, i1)
rtPath'(i1, i1)
認証点
p3=(j1+1, r(j1+1))
p2=(j1, s(j1))
p1=(j1, r(j1))
j1+1
j1
B
i0
i1

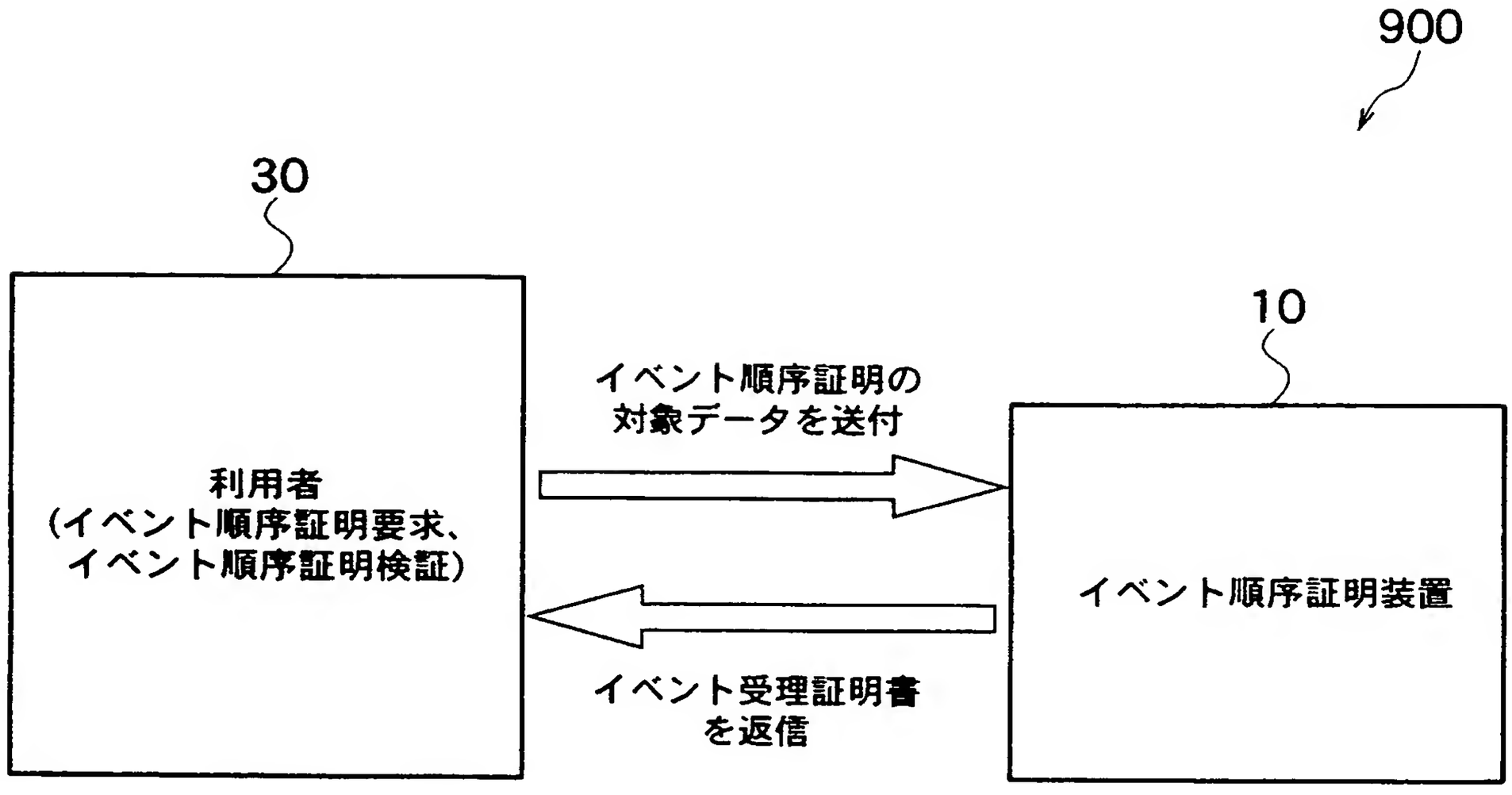
[図84]


ST2(4) = {i1}
i1
$2^{k(4)}$
ST2(3)
$2^{k(3)}$
i0のi1による認証点
p3=(j1+1, r(j1+1))
p2=(j1, s(j1))
p1=(j1, r(j1))
B
ST3
(=ST2(2))
i0
$2^{k(2)}$
ST2(1)
$2^{k(1)}$
j1+1
j1
k(1) >k(2) >k(3) >k(4) = 0

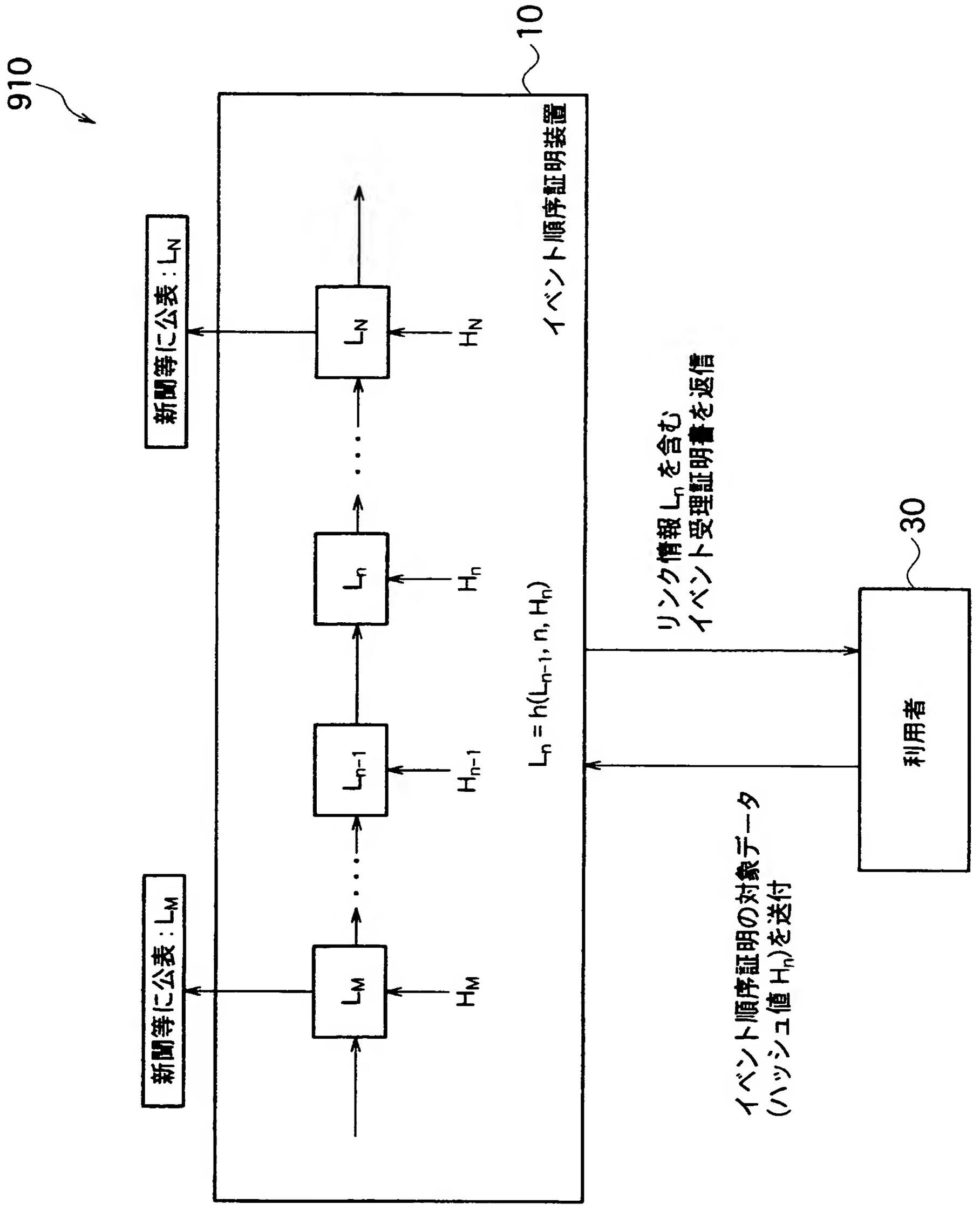
[図85]


ST2(4) = { i1 }
i1
2k(4)
ST2(3)
2k(3)
i0 の i1 による認証点
p3=(j1+1, r(j1+1))
p1=(j1, r(j1))
p2=(j1, s(j1))
B
ST3
(=ST2(2))
i0
2k(2)
ST2(1)
2k(1)
j1+1
j1
k(1) >k(2) >k(3) >k(4) = 0

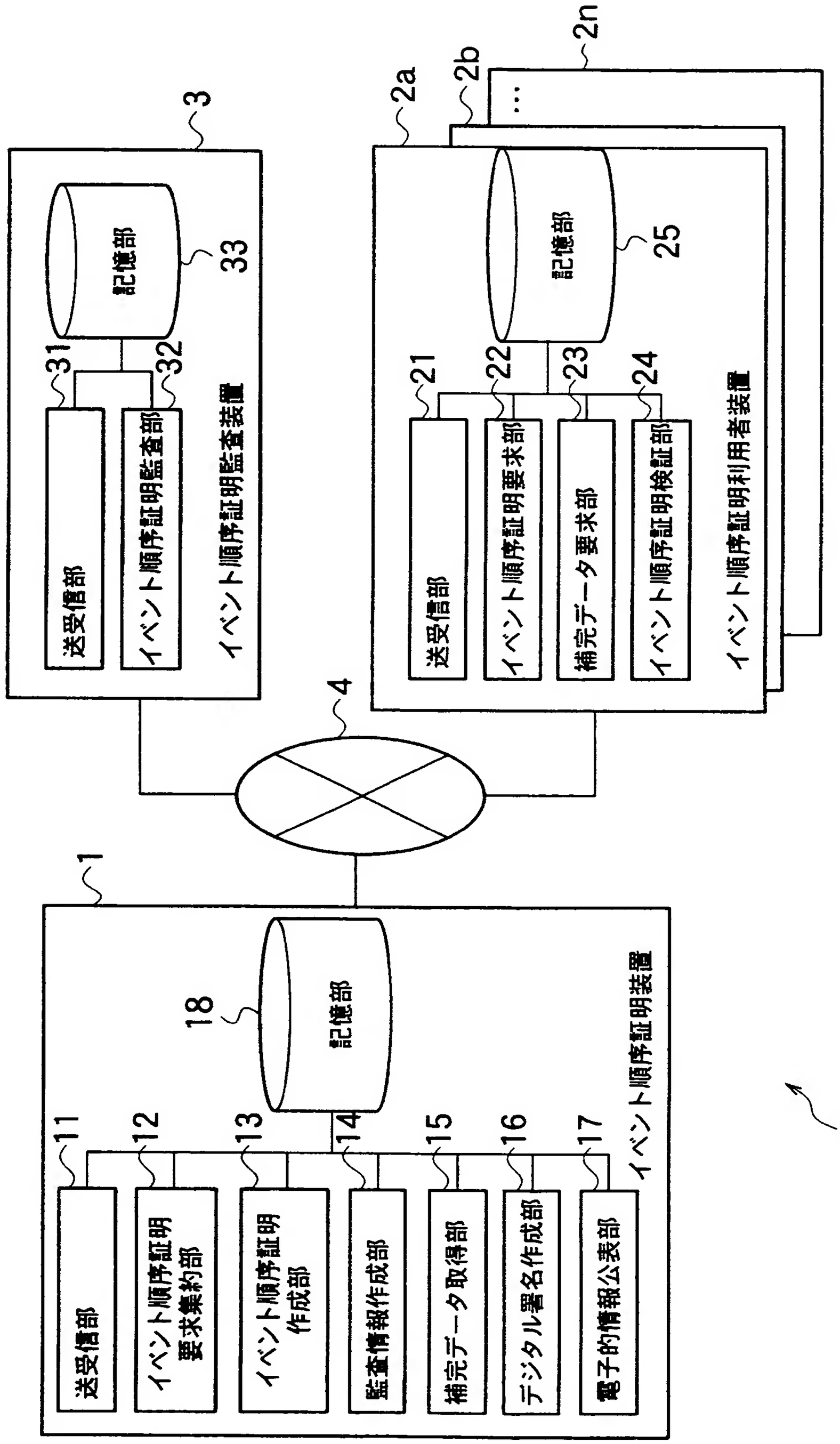
[図1]


900
30
利用者
(イベント順序証明要求、
イベント順序証明検証)
イベント順序証明の
対象データを送付
イベント受理証明書
を返信
10
イベント順序証明装置

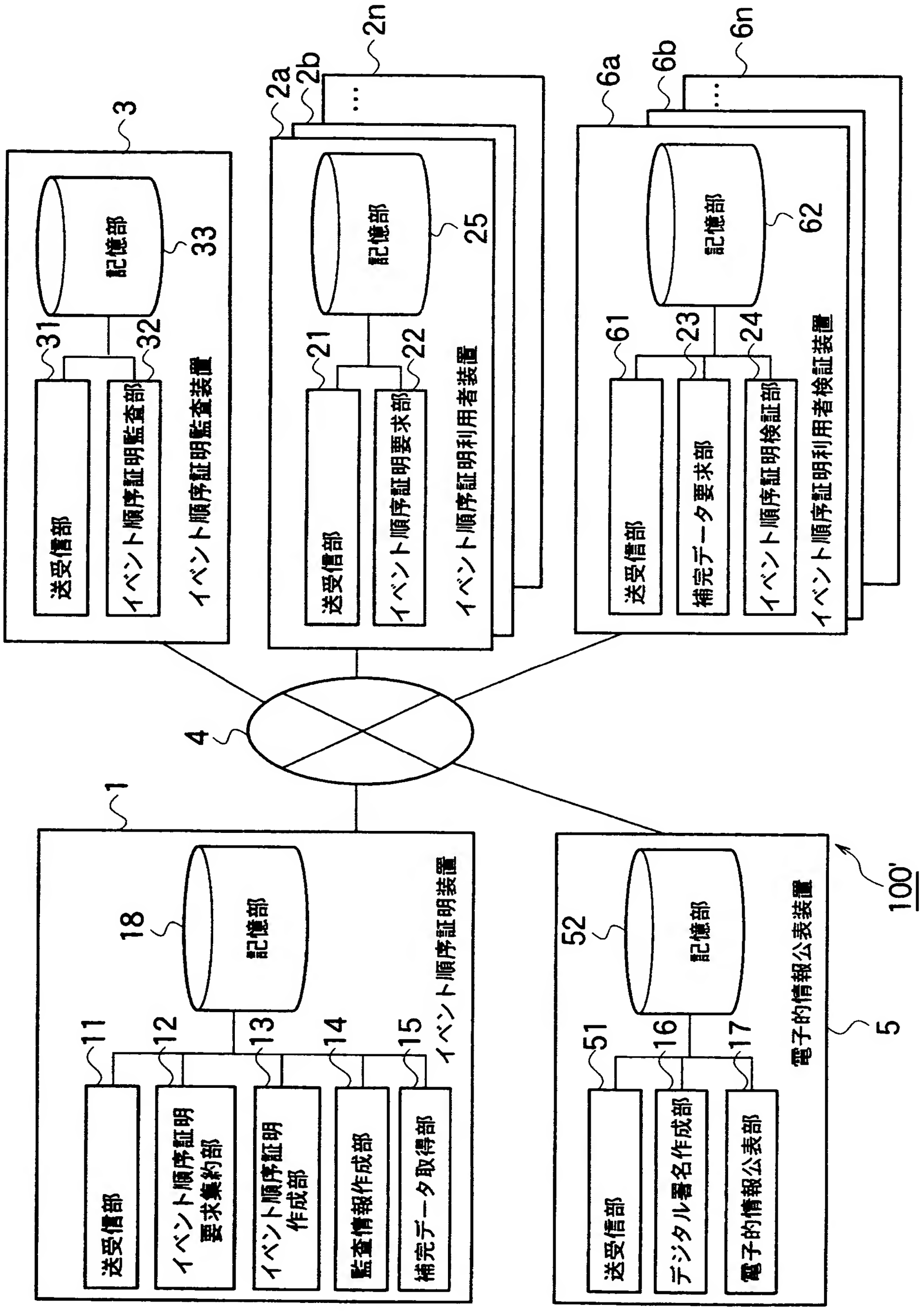
[図2]


910
10
新聞等に公表：L_N
L_N
H_N
L_n
H_n
L_{n-1}
H_{n-1}
新聞等に公表：L_M
L_M
H_M
イベント順序証明装置
$L_n = h(L_{n-1}, n, H_n)$
リンク情報L_nを含む
イベント受理証明書を返信
30
利用者
イベント順序証明の対象データ
(ハッシュ値H_n)を送付

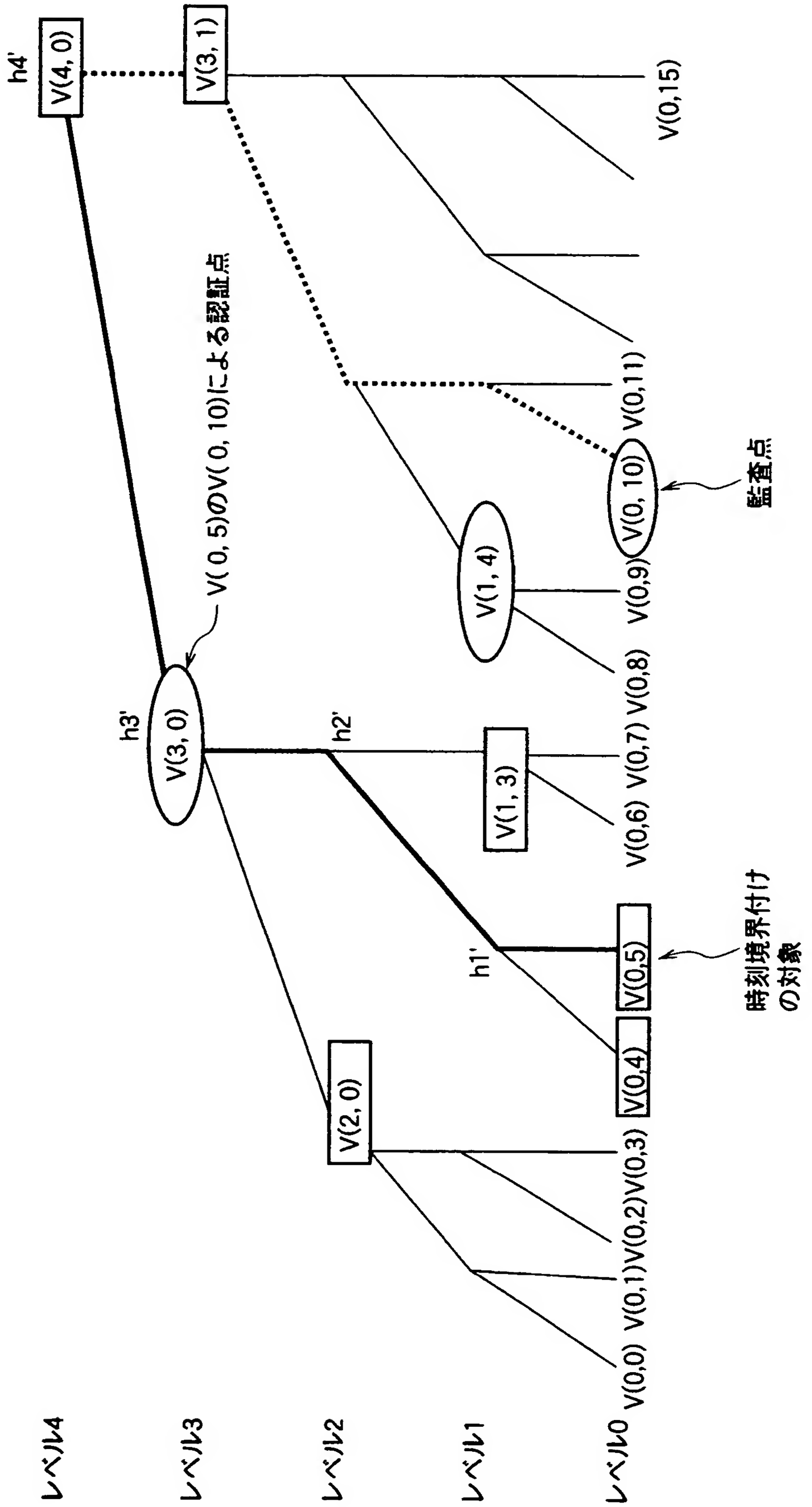
[図3]


100
イベント順序証明装置
11 送受信部
12 イベント順序証明要求集約部
13 イベント順序証明作成部
14 監査情報作成部
15 補完データ取得部
16 デジタル署名作成部
17 電子的情報公表部
18 記憶部
1
4
イベント順序証明監査装置
31 送受信部
32 イベント順序証明監査部
33 記憶部
3
イベント順序証明利用者装置
21 送受信部
22 イベント順序証明要求部
23 補完データ要求部
24 イベント順序証明検証部
25 記憶部
2a
2b
2n

[図4]


1
送受信部
イベント順序証明要求部
イベント順序証明作成部
監査情報作成部
補完データ取得部
記憶部
18
11
12
13
14
15
イベント順序証明装置
4
3
送受信部
イベント順序証明監査部
イベント順序証明監査装置
31
32
33
2a
2b
2n
送受信部
イベント順序証明要求部
イベント順序証明利用者装置
21
22
25
6a
6b
6n
送受信部
補完データ要求部
イベント順序証明検証部
イベント順序証明利用者検証装置
61
23
24
62
5
電子的情報公表装置
52
51
16
17
デジタル署名作成部
電子的情報公表部
記憶部
100'

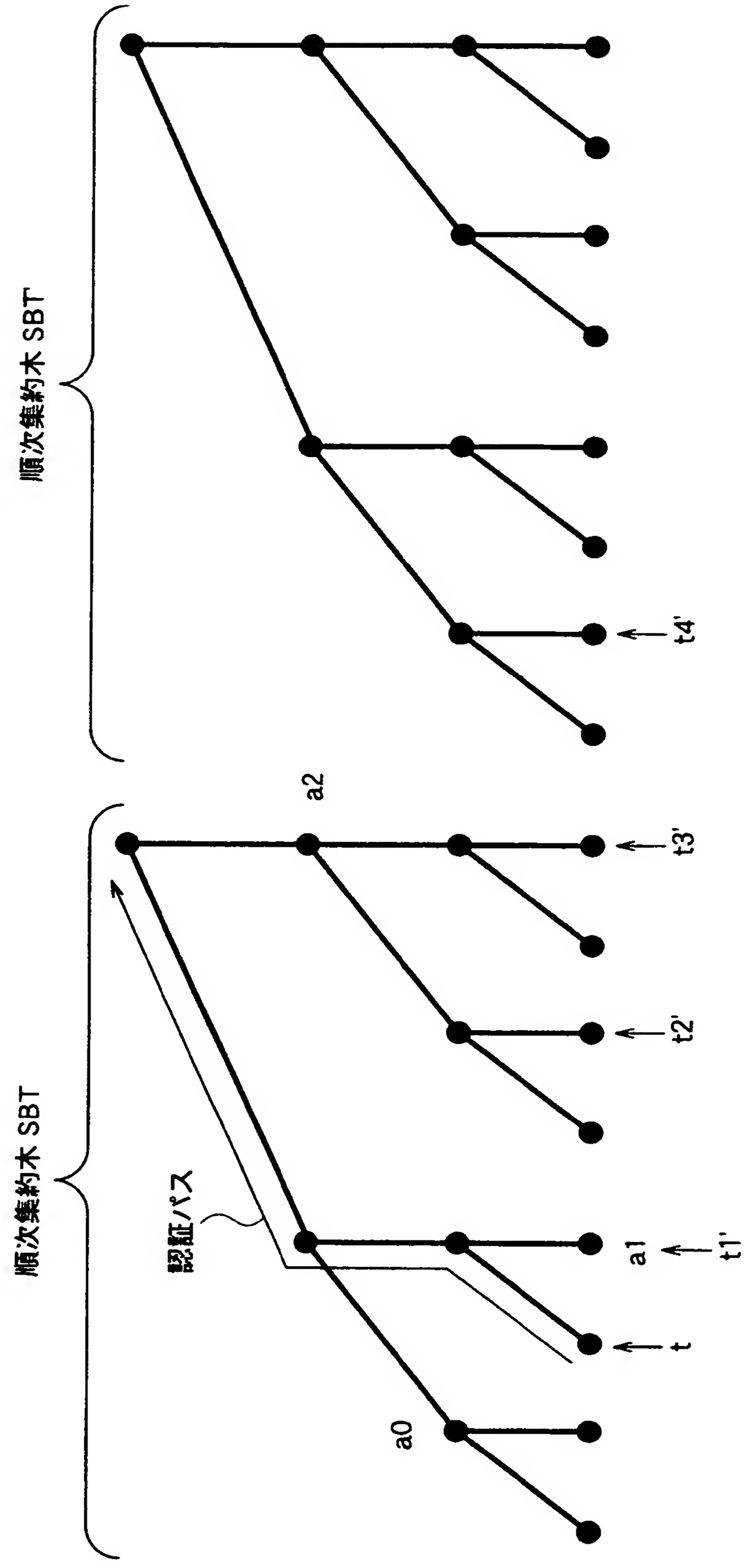
[図5]


h4'
V(4, 0)
V(3, 1)
V(0,15)
V(0, 5)∅V(0, 10)による認証点
h3'
V(3, 0)
h2'
V(1, 4)
V(1, 3)
V(0,11)
V(0, 10)
V(0,9)
V(0,8)
V(0,7)
V(0,6)
監査点
h1'
V(0,5)
V(0,4)
時刻境界付け
の対象
V(2, 0)
V(0,3)
V(0,2)
V(0,1)
V(0,0)
レベル4
レベル3
レベル2
レベル1
レベル0

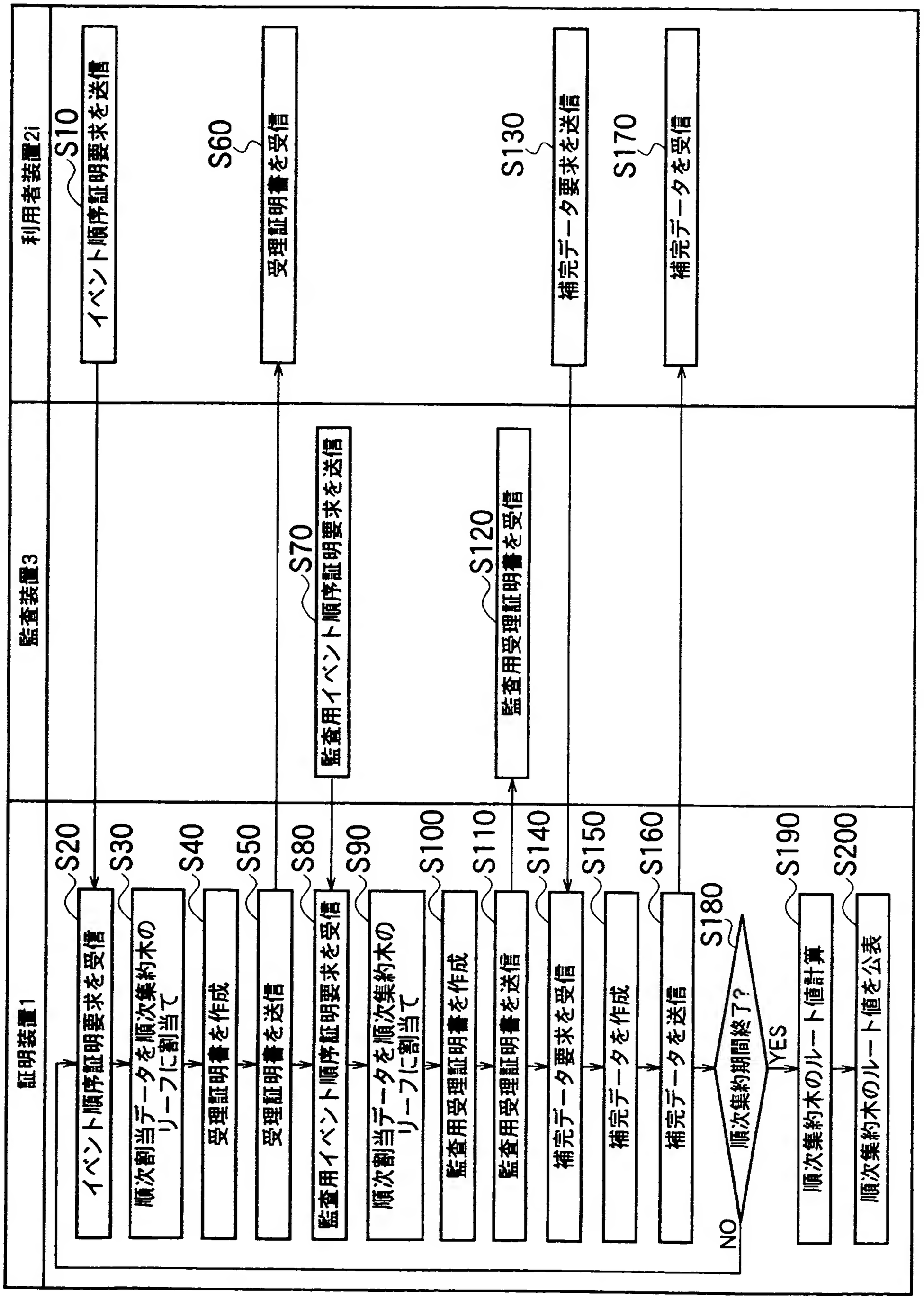
[図6]

項目	記号	必須
元デジタルデータ	y	○
順次割当データ	z	○
順次集約木番号	n	○
順次集約木リーフ番号	i	○
即時補完データ(位置情報、割当値)	HK	
デジタル署名	DS	

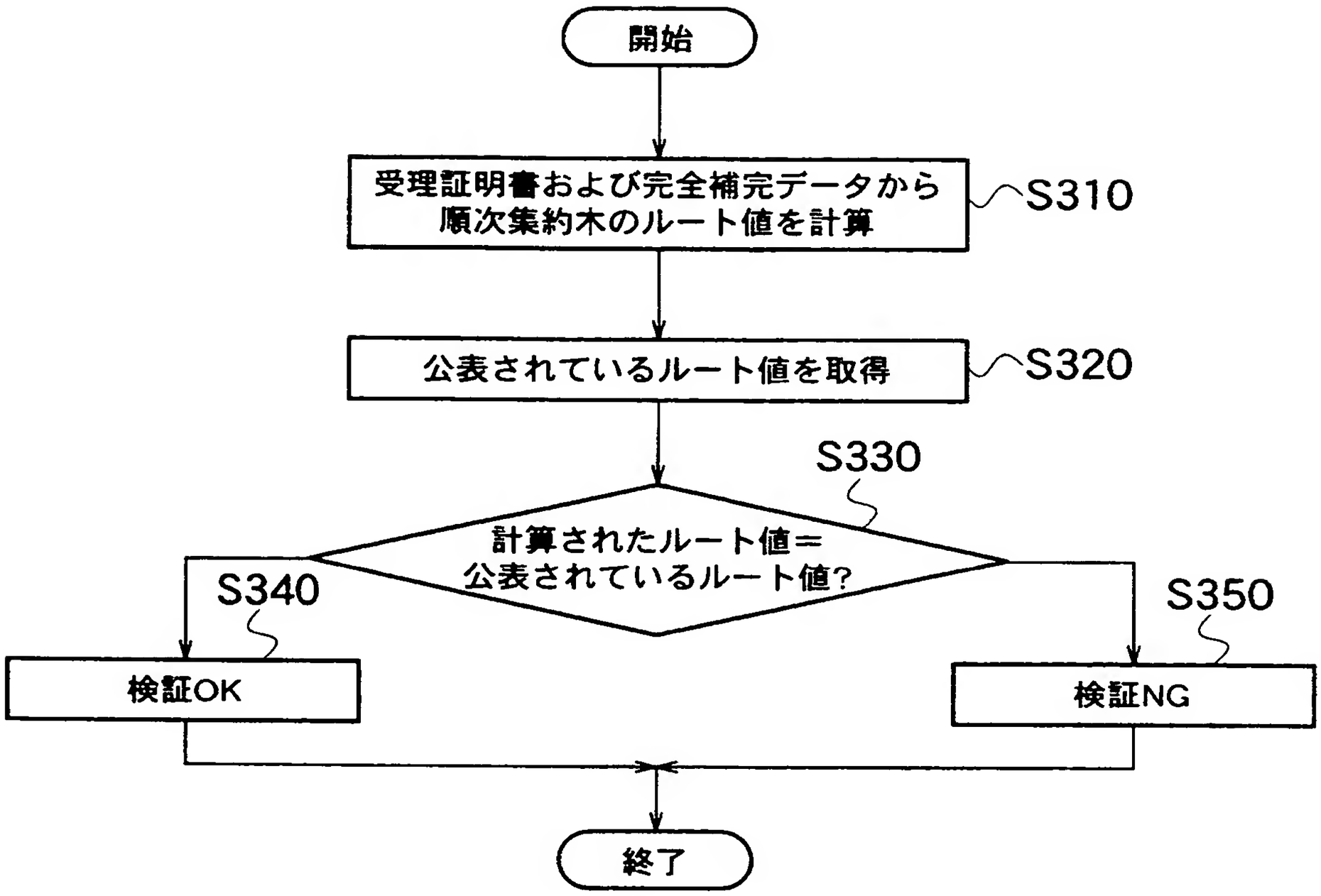
[図7]



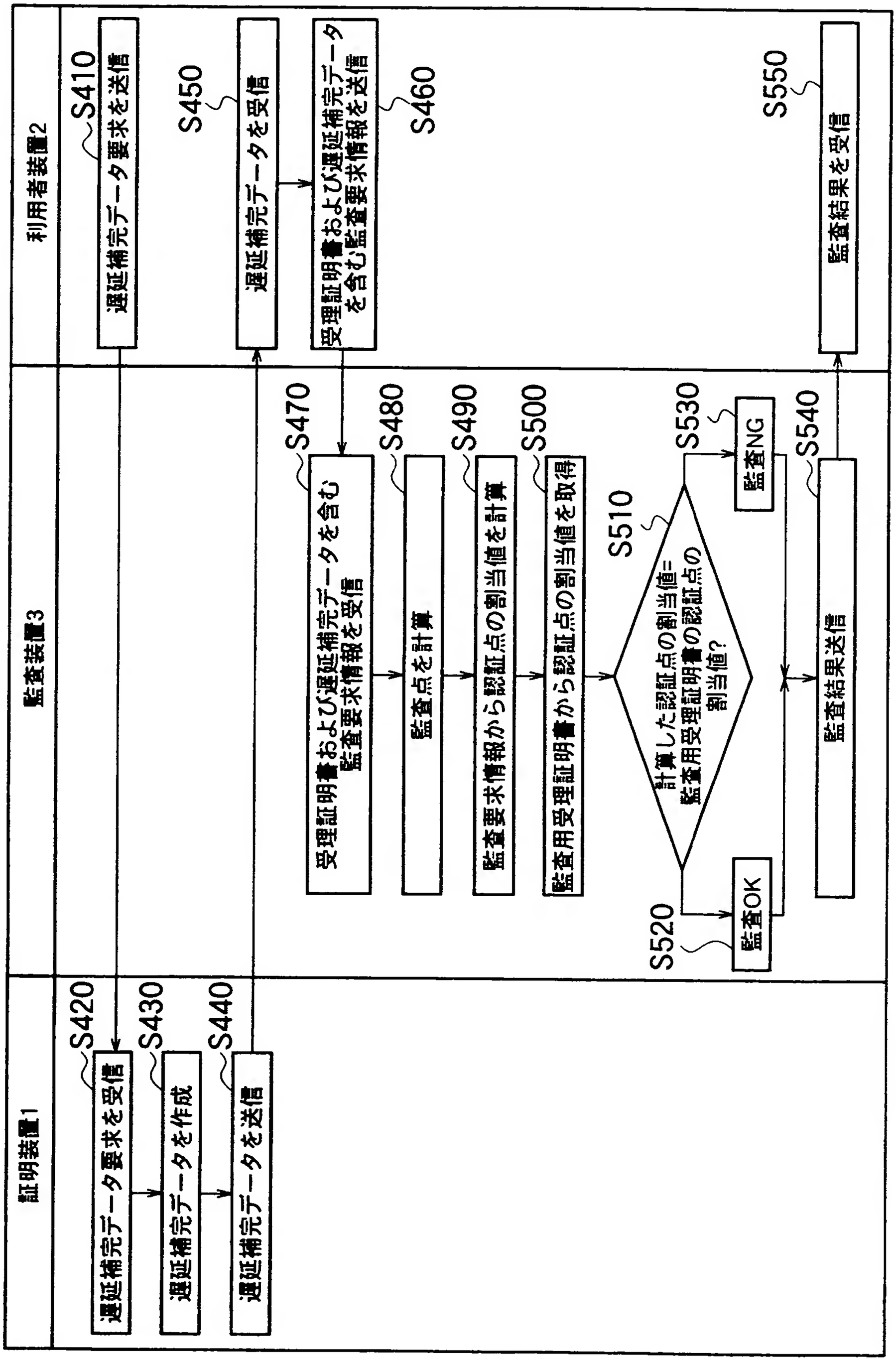
[図8]


利用者装置2i
監査装置3
証明装置1
S10
イベント順序証明要求を送信
S60
受理証明書を受信
S130
補完データ要求を送信
S170
補完データを受信
S70
監査用イベント順序証明要求を送信
S120
監査用受理証明書を受信
S20
イベント順序証明要求を受信
S30
順次割当データを順次集約木のリーフに割当て
S40
受理証明書を作成
S50
受理証明書を送信
S80
監査用イベント順序証明要求を受信
S90
順次割当データを順次集約木のリーフに割当て
S100
監査用受理証明書を作成
S110
監査用受理証明書を送信
S140
補完データ要求を受信
S150
補完データを作成
S160
補完データを送信
S180
順次集約期間終了?
NO
YES
S190
順次集約木のルート値計算
S200
順次集約木のルート値を公表

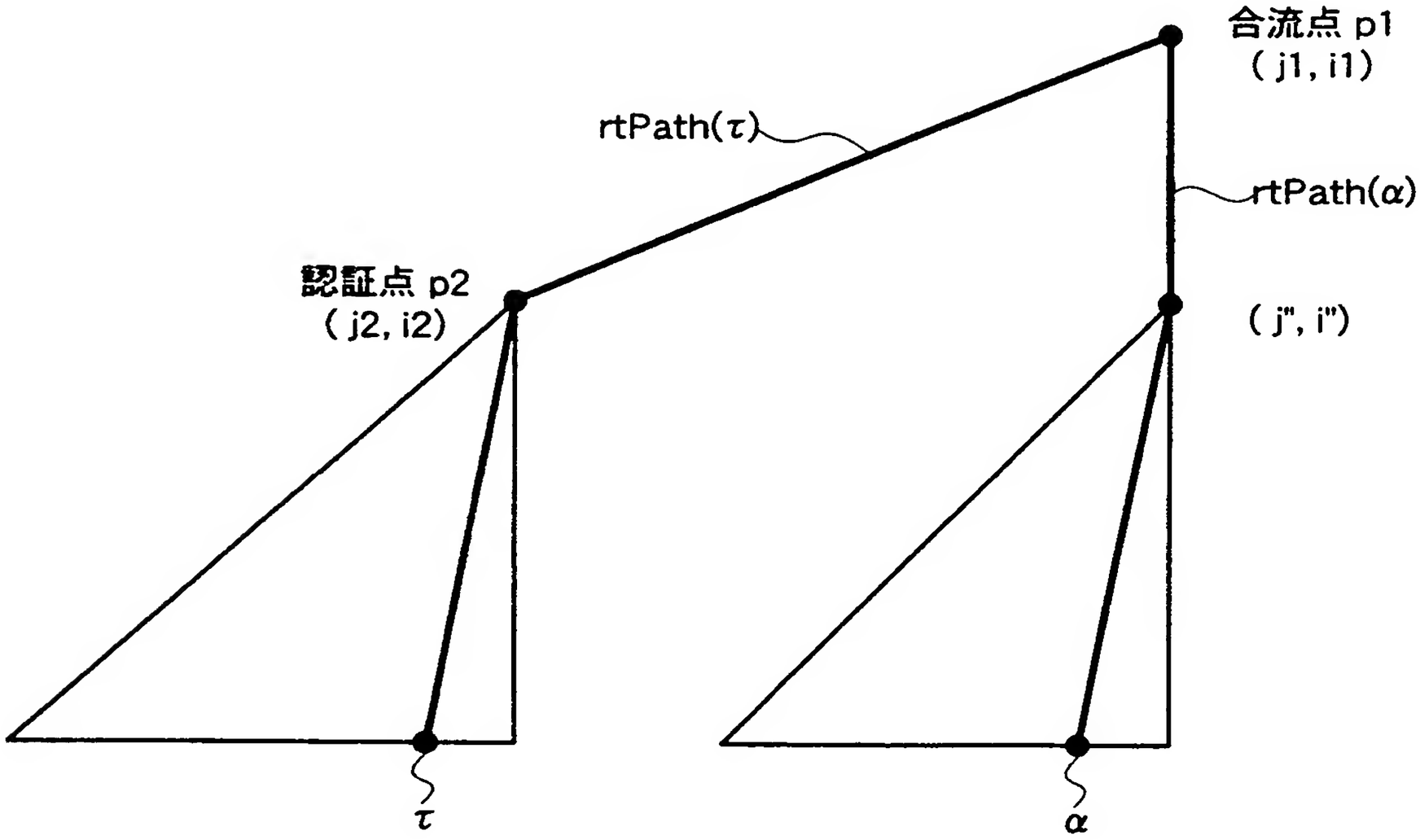
[図9]


開始
受理証明書および完全補完データから順次集約木のルート値を計算
S310
公表されているルート値を取得
S320
S330
計算されたルート値＝公表されているルート値?
S340
検証OK
S350
検証NG
終了

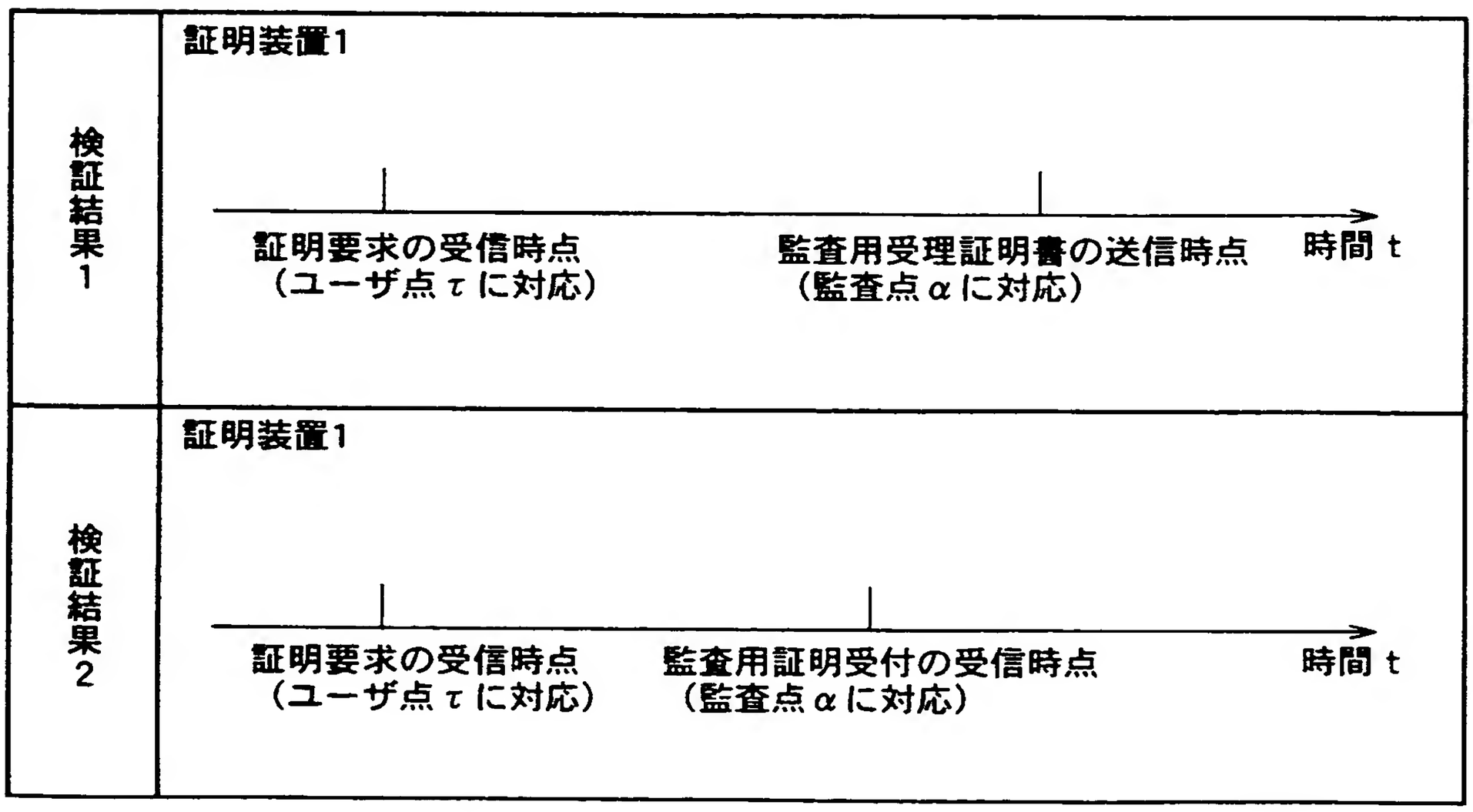
[図10]


利用者装置2
S410
遅延補完データ要求を送信
S450
遅延補完データを受信
S460
受理証明書および遅延補完データを含む監査要求情報を送信
S550
監査結果を受信
監査装置3
S470
受理証明書および遅延補完データを含む監査要求情報を受信
S480
監査点を計算
S490
監査要求情報から認証点の割当値を計算
S500
監査用受理証明書から認証点の割当値を取得
S510
計算した認証点の割当値＝監査用受理証明書の認証点の割当値？
S530
監査NG
S540
監査結果送信
S520
監査OK
証明装置1
S420
遅延補完データ要求を受信
S430
遅延補完データを作成
S440
遅延補完データを送信

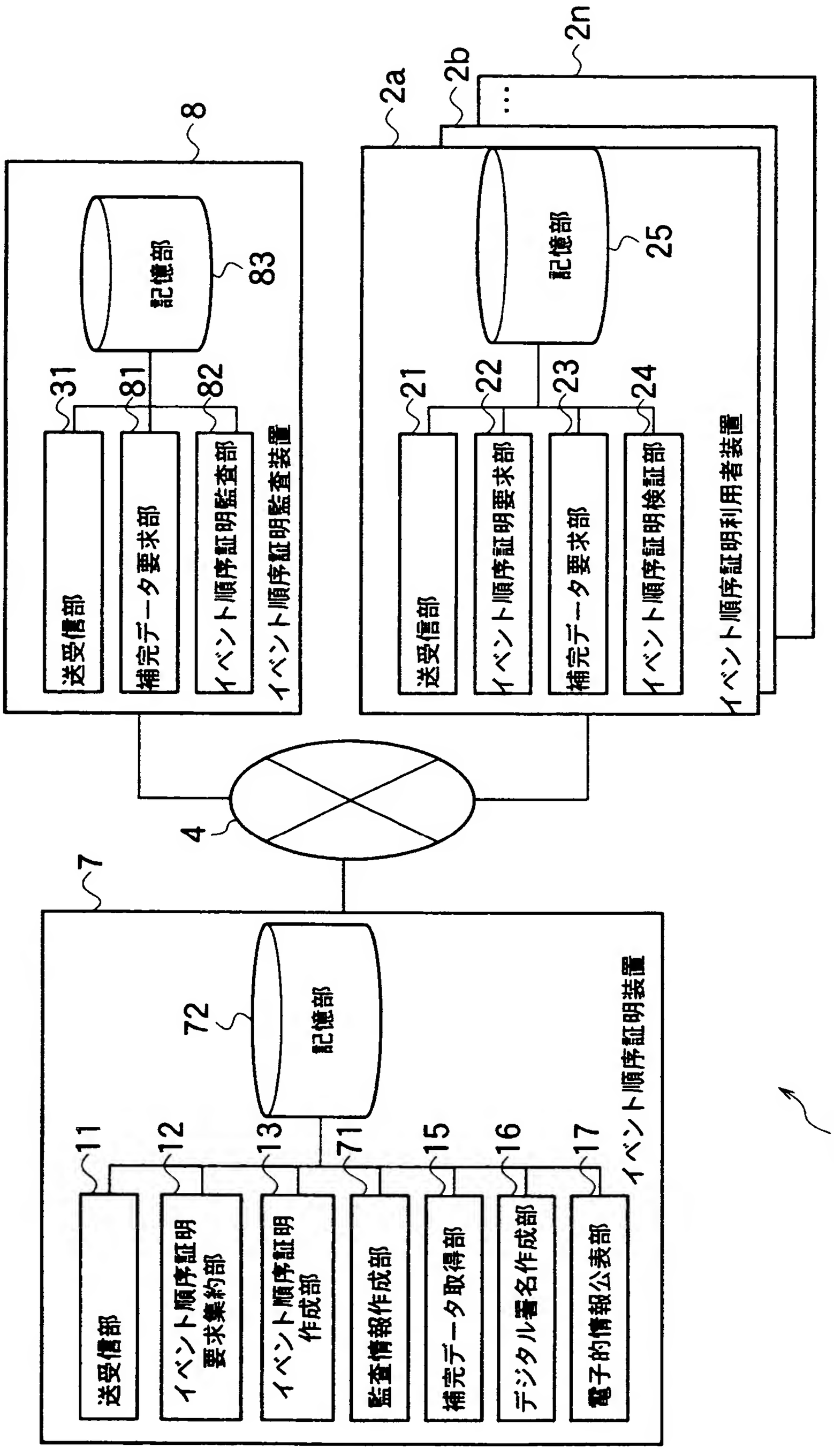
[図11]


合流点 p1
(j1, i1)
rtPath(τ)
rtPath(α)
認証点 p2
(j2, i2)
(j", i")
τ
α

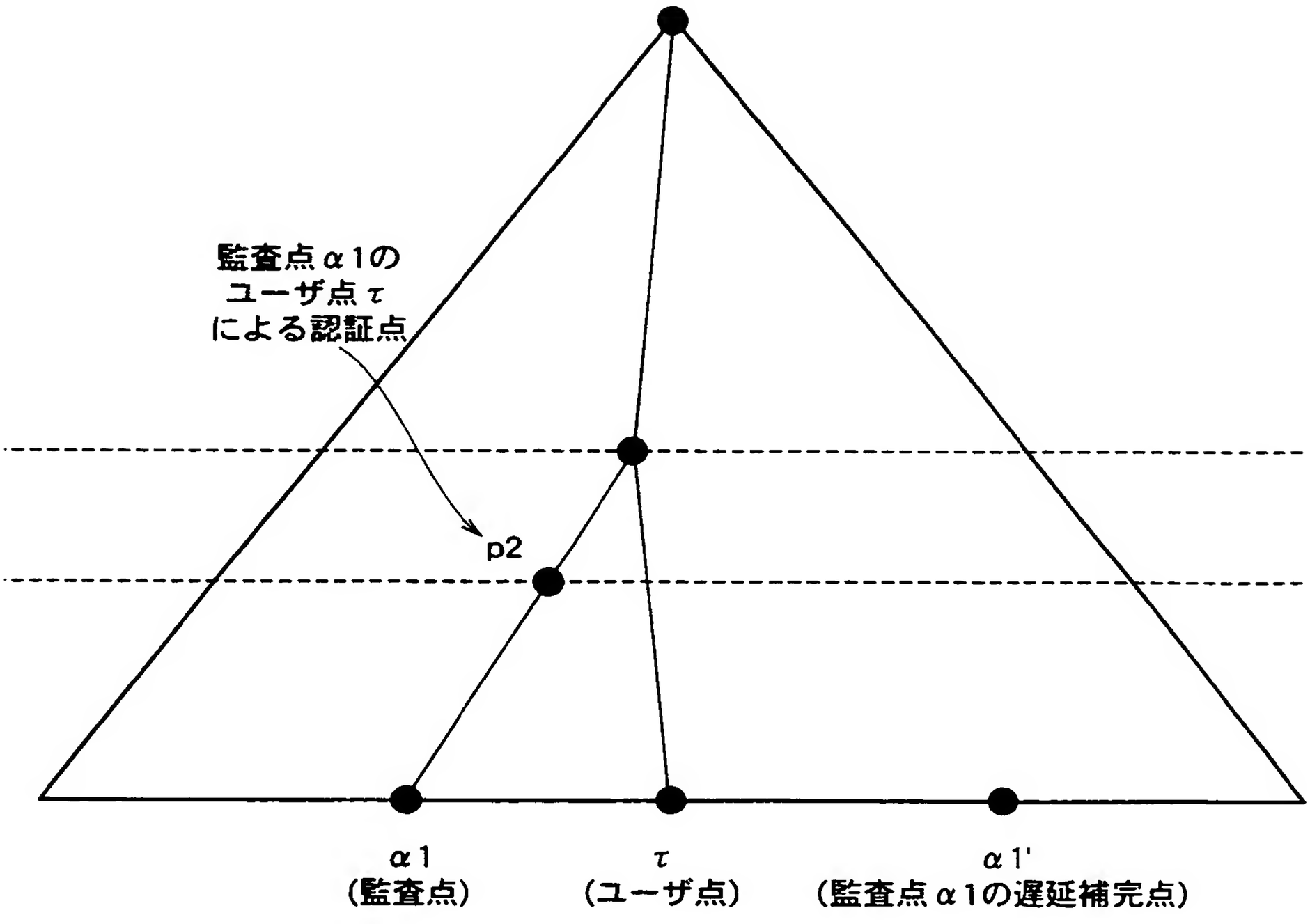
[図12]


検証結果1
証明装置1
証明要求の受信時点
（ユーザ点τに対応）
監査用受理証明書の送信時点
（監査点αに対応）
時間 t
検証結果2
証明装置1
証明要求の受信時点
（ユーザ点τに対応）
監査用証明受付の受信時点
（監査点αに対応）
時間 t

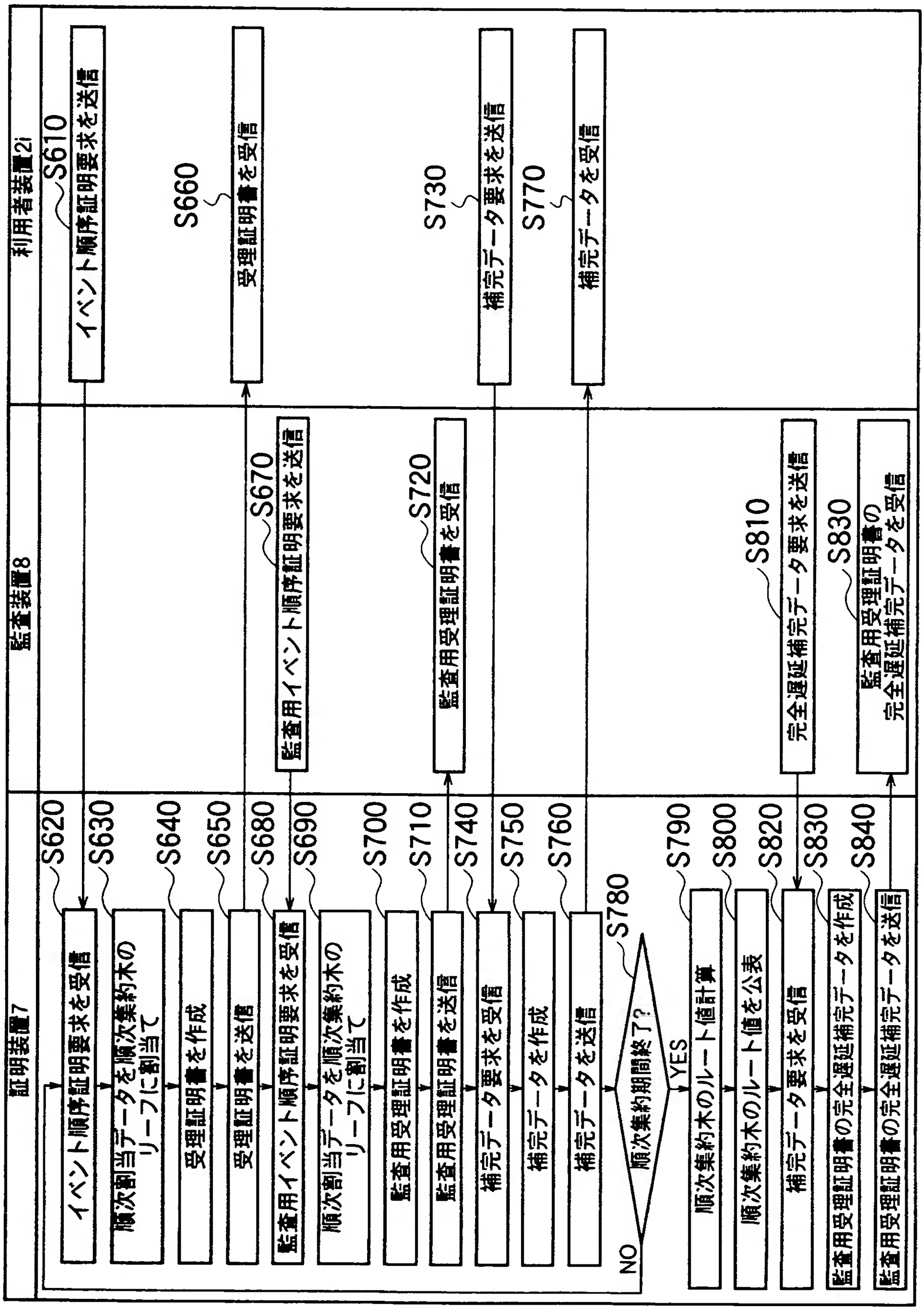
[図13]


200
イベント順序証明装置
送受信部
イベント順序証明要求集約部
イベント順序証明作成部
監査情報作成部
補完データ取得部
デジタル署名作成部
電子的情報公表部
記憶部
11
12
13
71
72
15
16
17
7
4
イベント順序証明監査装置
送受信部
補完データ要求部
イベント順序証明監査部
記憶部
31
81
82
83
8
イベント順序証明利用者装置
送受信部
イベント順序証明要求部
補完データ要求部
イベント順序証明検証部
記憶部
21
22
23
24
25
2a
2b
2n

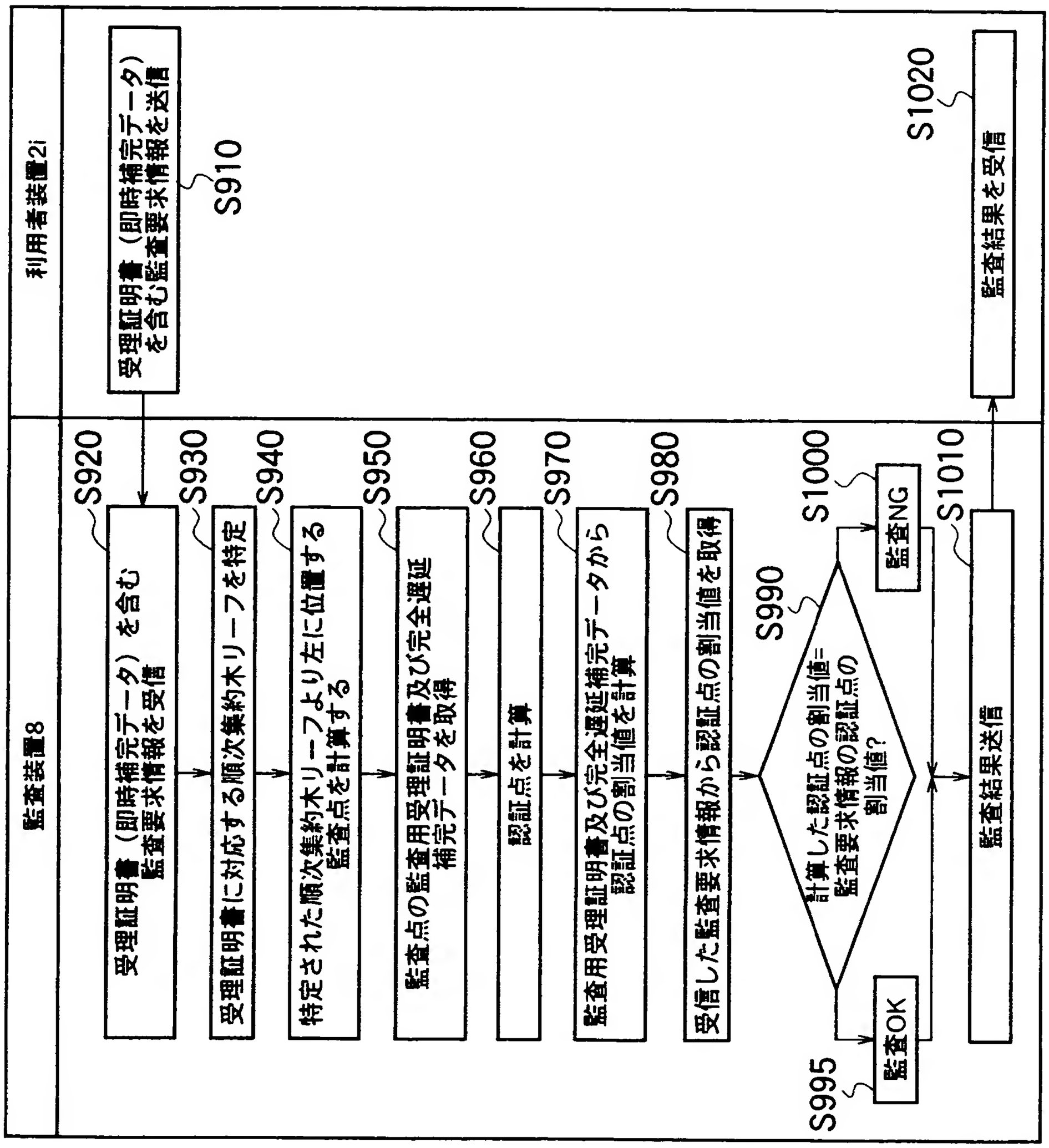
[図14]



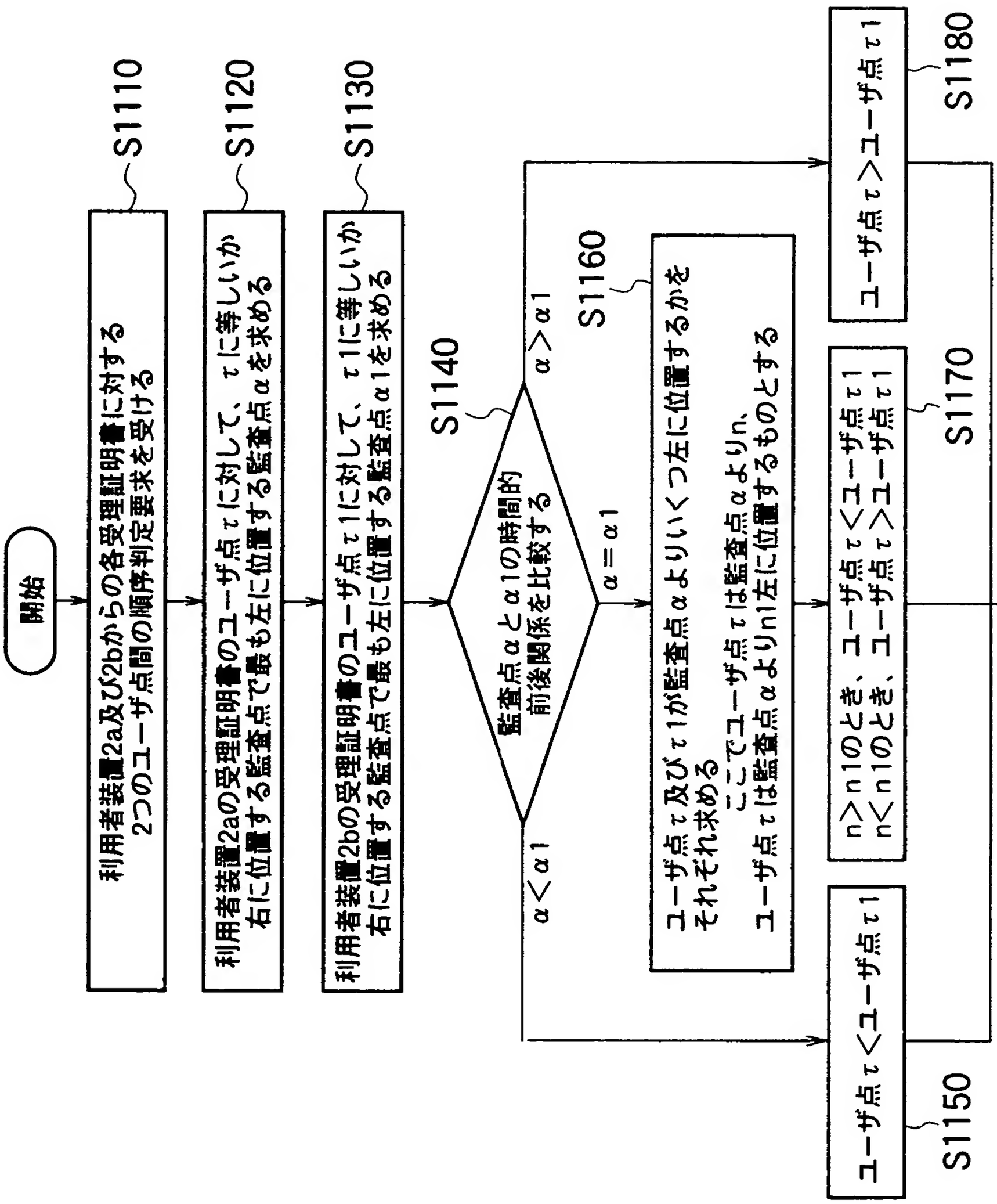
[図15]


利用者装置2i
監査装置8
証明装置7
S610
イベント順序証明要求を送信
S660
受理証明書を受信
S730
補完データ要求を送信
S770
補完データを受信
S620
イベント順序証明要求を受信
S630
順次割当データを順次集約木のリーフに割当て
S640
受理証明書を作成
S650
受理証明書を送信
S670
監査用イベント順序証明要求を送信
S680
監査用イベント順序証明要求を受信
S690
順次割当データを順次集約木のリーフに割当て
S700
監査用受理証明書を作成
S710
監査用受理証明書を送信
S720
監査用受理証明書を受信
S740
補完データ要求を受信
S750
補完データを作成
S760
補完データを送信
S780
順次集約期間終了?
NO
YES
S790
順次集約木のルート値計算
S800
順次集約木のルート値を公表
S810
完全遅延補完データ要求を送信
S820
補完データ要求を受信
S830
監査用受理証明書の
完全遅延補完データを受信
S840
監査用受理証明書の完全遅延補完データを作成
S850
監査用受理証明書の完全遅延補完データを送信

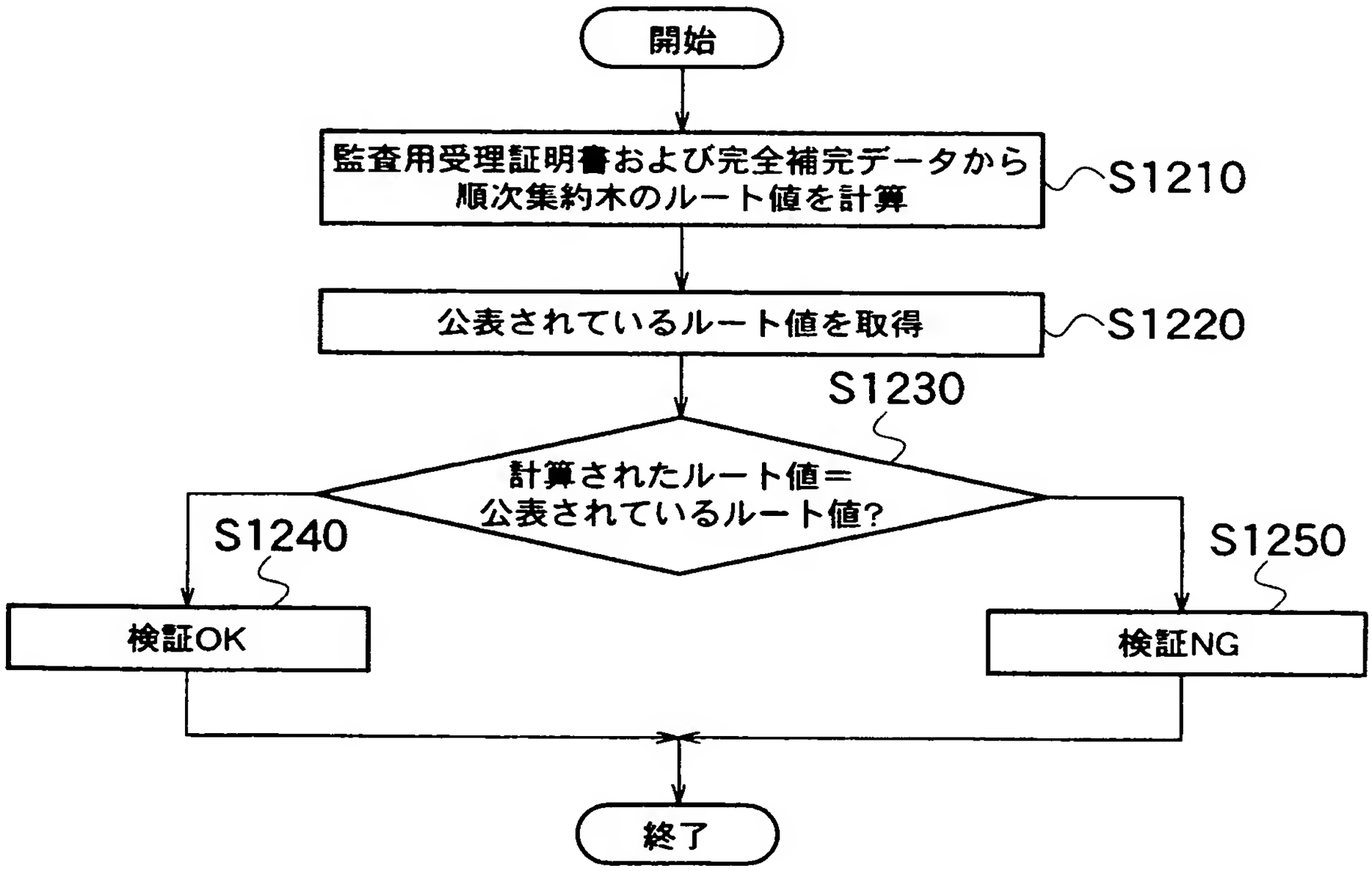
[図16]


利用者装置2i
監査装置8
受理証明書（即時補完データ）
監査要求情報を送信
S910
受理証明書（即時補完データ）を含む
監査要求情報を受信
S920
受理証明書に対応する順次集約木リーフを特定
S930
特定された順次集約木リーフより左に位置する
監査点を計算する
S940
監査点の監査用受理証明書及び完全遅延
補完データを取得
S950
認証点を計算
S960
監査用受理証明書及び完全遅延補完データから
認証点の割当値を計算
S970
受信した監査要求情報から認証点の割当値を取得
S980
計算した認証点の割当値＝
監査要求情報の認証点の
割当値?
S990
監査NG
S1000
監査OK
S995
監査結果送信
S1010
監査結果を受信
S1020

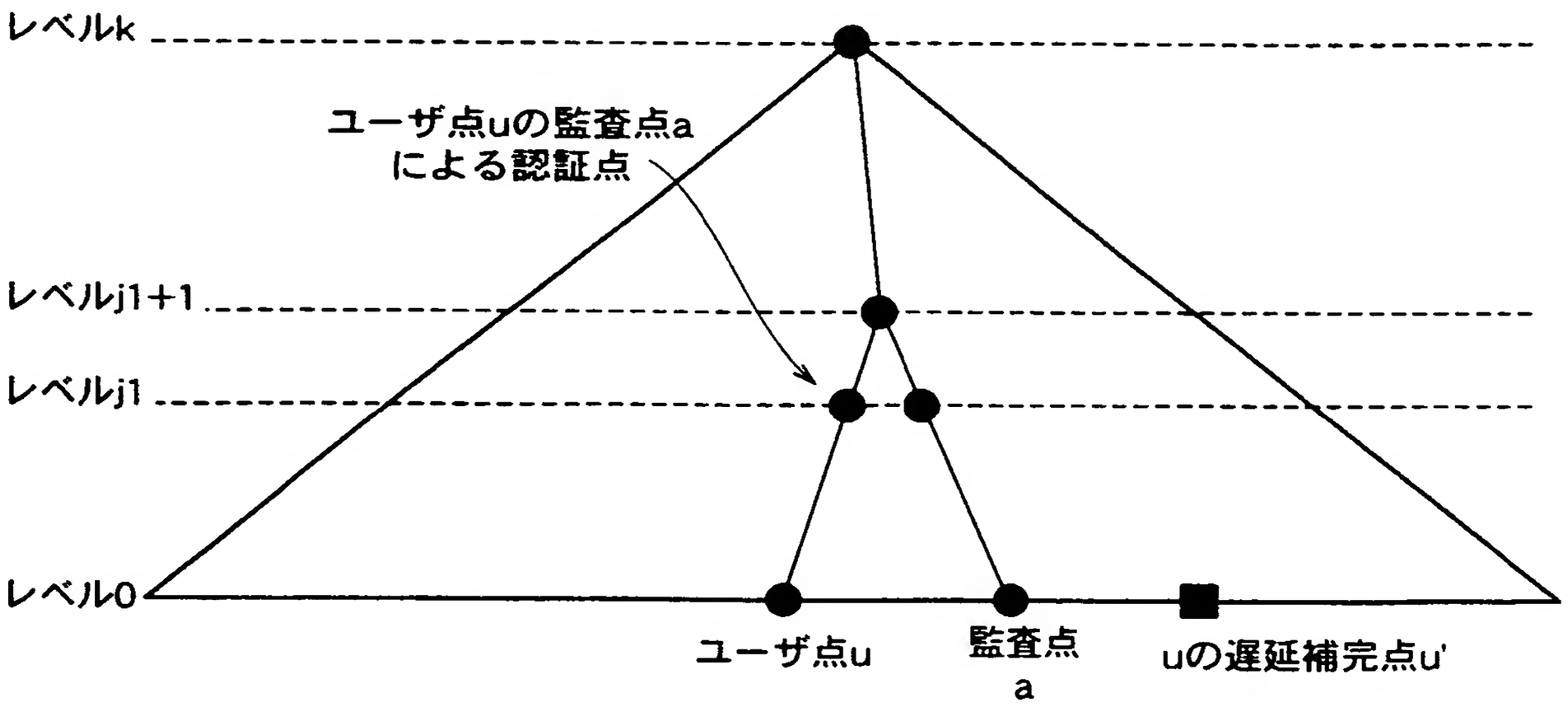
[図17]


開始
利用者装置2a及び2bからの各受理証明書に対する2つのユーザ点間の順序判定要求を受ける
S1110
利用者装置2aの受理証明書のユーザ点τに対して、τに等しいか右に位置する監査点で最も左に位置する監査点αを求める
S1120
利用者装置2bの受理証明書のユーザ点τ1に対して、τ1に等しいか右に位置する監査点で最も左に位置する監査点α1を求める
S1130
S1140
監査点αとα1の時間的前後関係を比較する
α＜α1
α＝α1
α＞α1
S1160
ユーザ点τ及びτ1が監査点αよりいくつ左に位置するかをそれぞれ求める
ここでユーザ点τは監査点αよりn、ユーザ点τは監査点αよりn1左に位置するものとする
S1170
n＞n1のとき、ユーザ点τ＜ユーザ点τ1
n＜n1のとき、ユーザ点τ＞ユーザ点τ1
ユーザ点τ＜ユーザ点τ1
S1150
ユーザ点τ＞ユーザ点τ1
S1180
終了

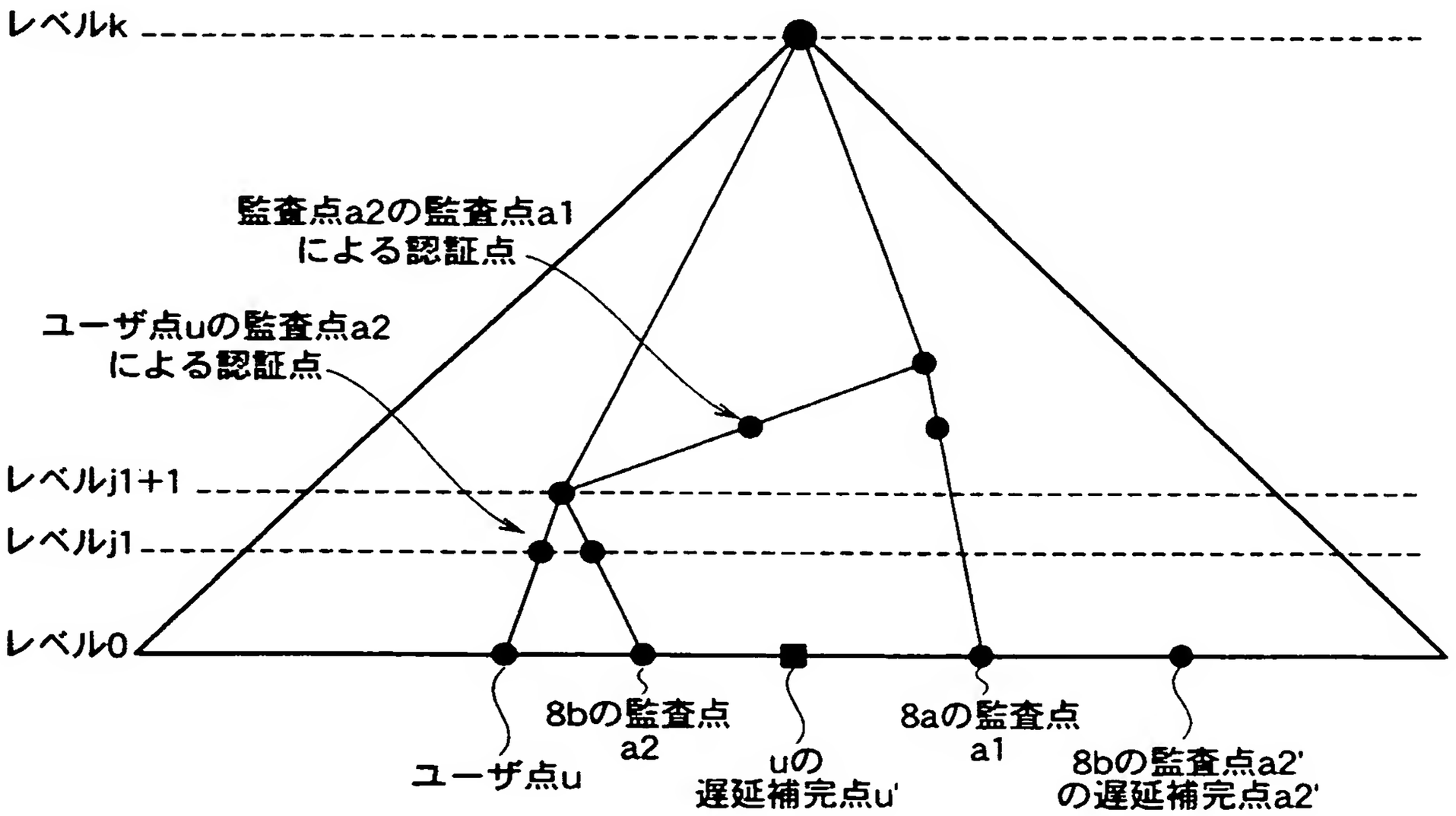
[図18]


開始
監査用受理証明書および完全補完データから順次集約木のルート値を計算
S1210
公表されているルート値を取得
S1220
S1230
計算されたルート値＝公表されているルート値?
S1240
検証OK
S1250
検証NG
終了

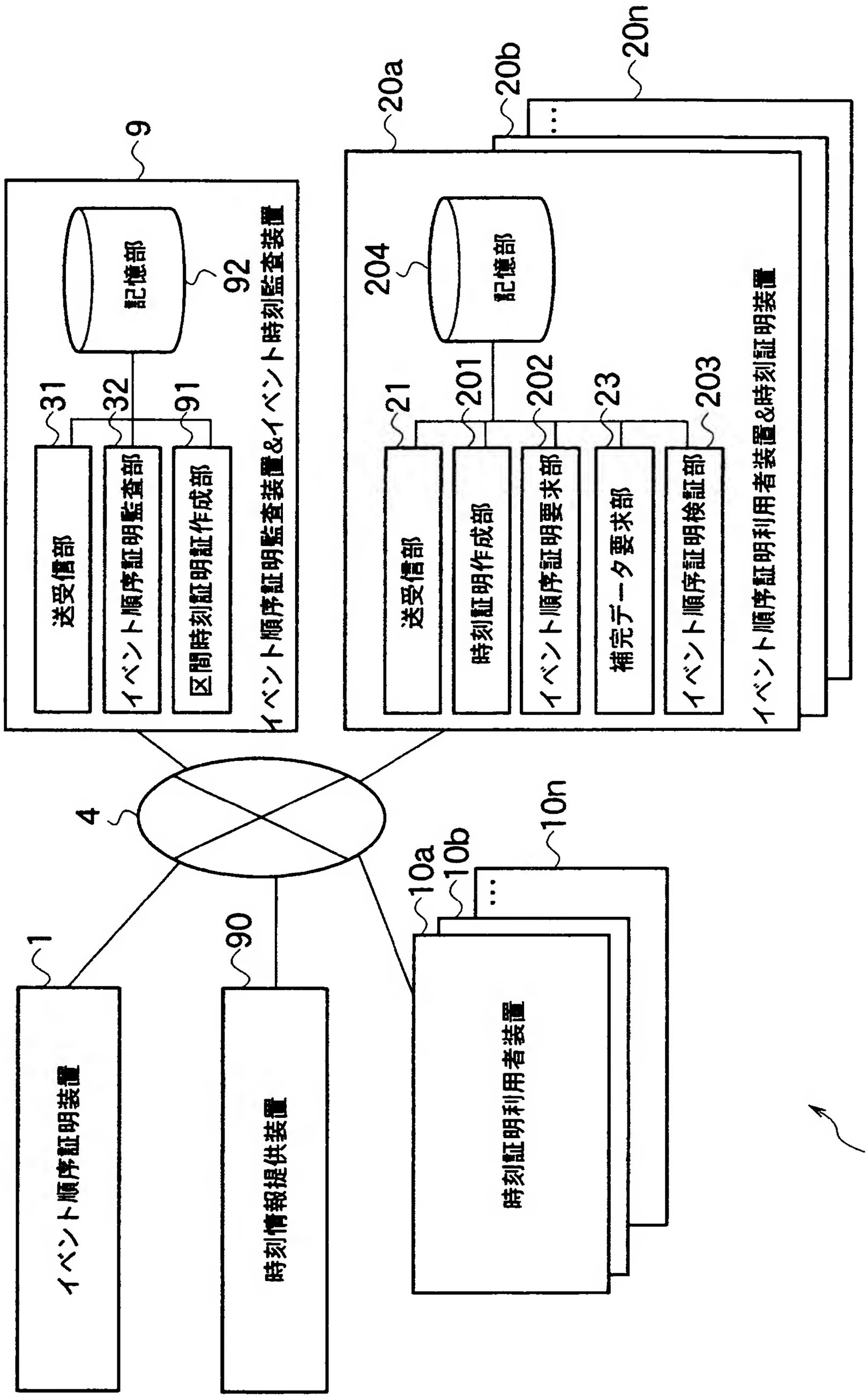
[図19]


レベルk
ユーザ点uの監査点a
による認証点
レベルj1+1
レベルj1
レベル0
ユーザ点u
監査点
a
uの遅延補完点u'

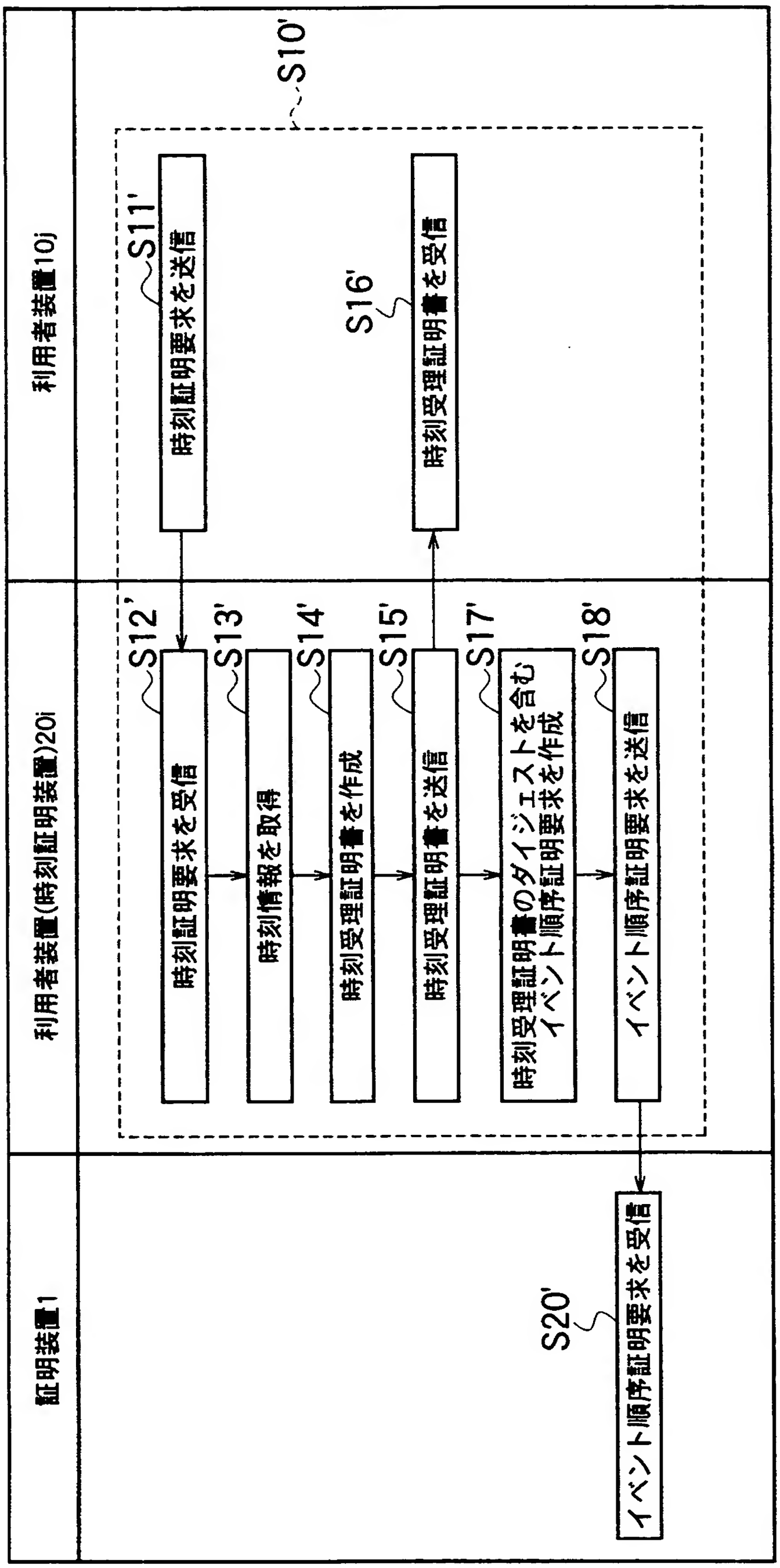
[図20]


レベルk
監査点a2の監査点a1
による認証点
ユーザ点uの監査点a2
による認証点
レベルj1+1
レベルj1
レベル0
8bの監査点
a2
8aの監査点
a1
ユーザ点u
uの
遅延補完点u'
8bの監査点a2'
の遅延補完点a2'

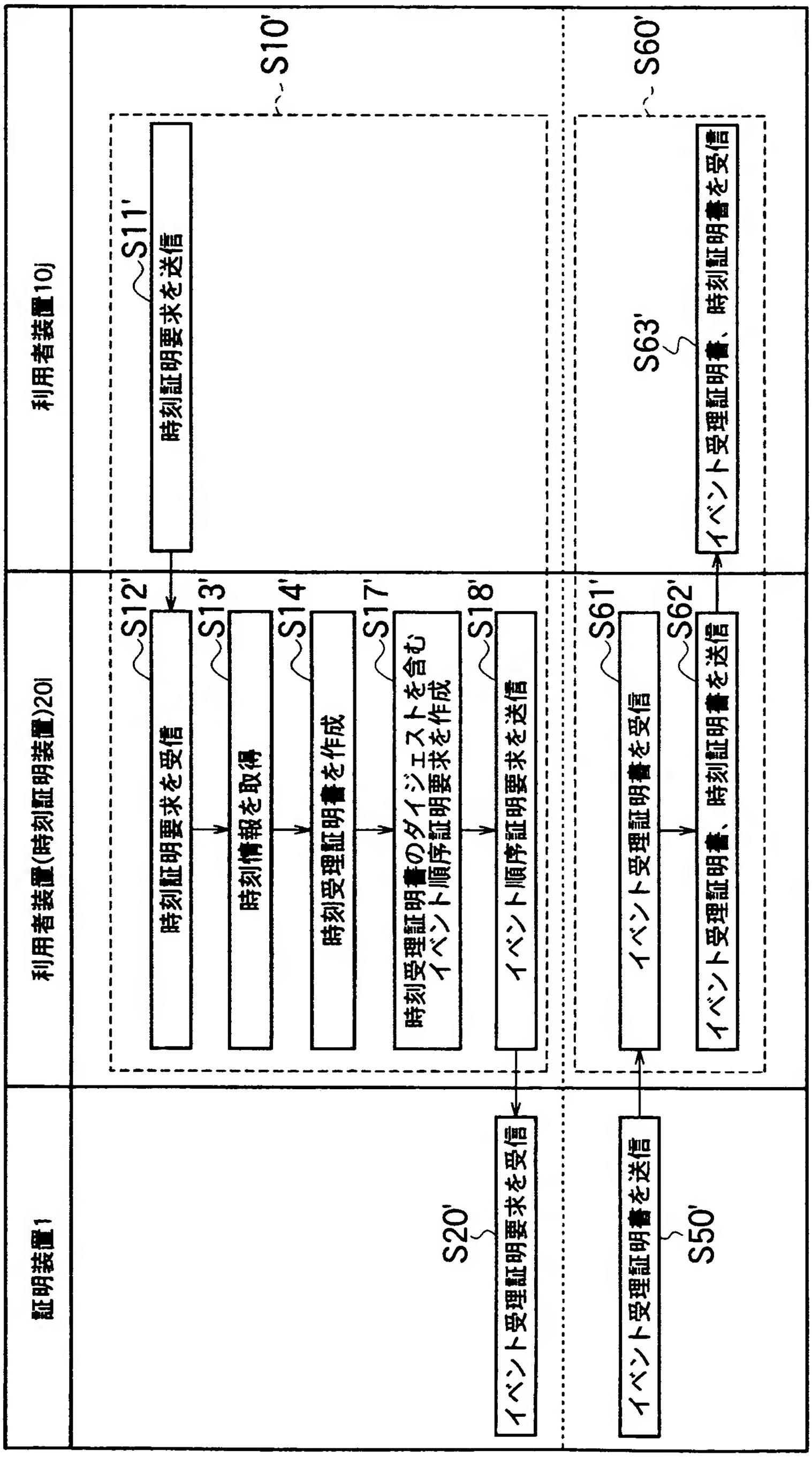
[図21]


300
イベント順序証明装置
1
時刻情報提供装置
90
時刻証明利用者装置
10a
10b
10n
4
イベント順序証明監査装置&イベント時刻証明装置
9
記憶部
92
送受信部
31
イベント順序証明監査部
32
区間時刻証明証作成部
91
イベント順序証明利用者装置&時刻証明装置
20a
20b
20n
記憶部
204
送受信部
21
時刻証明作成部
201
イベント順序証明要求部
202
補完データ要求部
23
イベント順序証明検証部
203

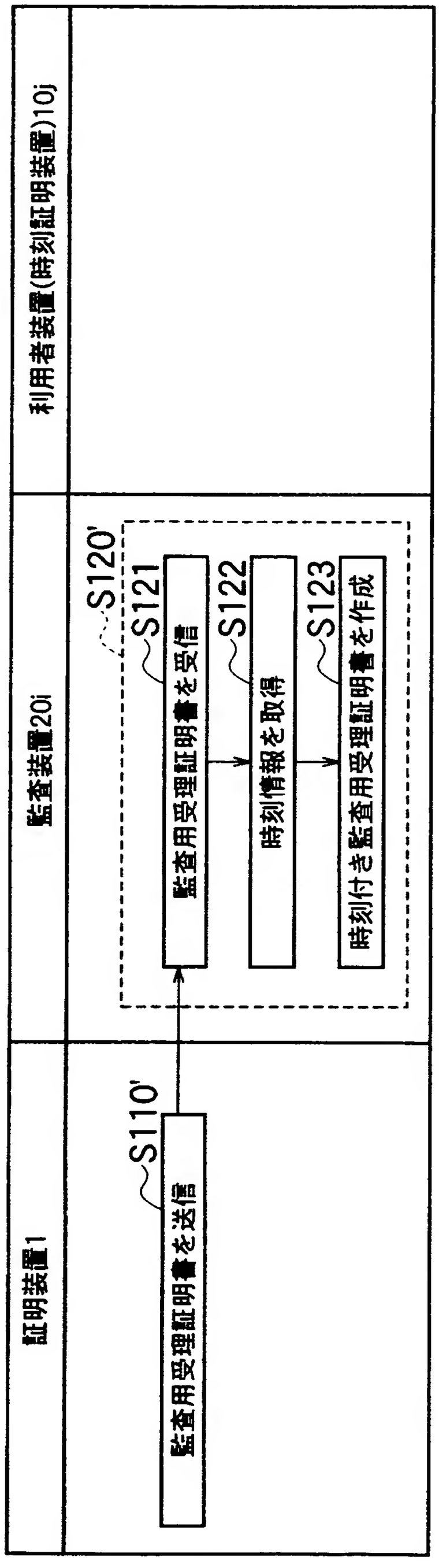
[図22]


利用者装置10j
S10'
S11'
時刻証明要求を送信
S16'
時刻受理証明書を受信
利用者装置(時刻証明装置)20i
S12'
時刻証明要求を受信
S13'
時刻情報を取得
S14'
時刻受理証明書を作成
S15'
時刻受理証明書を送信
S17'
時刻受理証明書のダイジェストを含む
イベント順序証明要求を作成
S18'
イベント順序証明要求を送信
証明装置1
S20'
イベント順序証明要求を受信

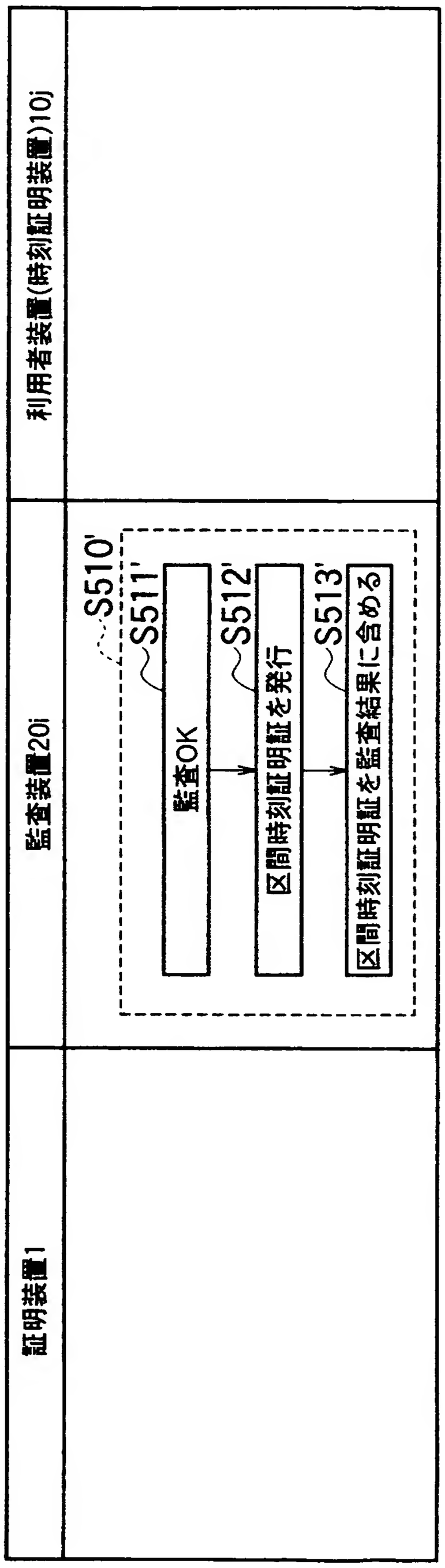
[図23]


利用者装置10j
証明装置1
利用者装置(時刻証明装置)20i
S10'
S11'
時刻証明要求を送信
S12'
時刻証明要求を受信
S13'
時刻情報を取得
S14'
時刻受理証明書を作成
S17'
時刻受理証明書のダイジェストを含む
イベント順序証明要求を作成
S18'
イベント順序証明要求を送信
S20'
イベント受理証明要求を受信
S60'
S63'
イベント受理証明書、時刻証明書を受信
S61'
イベント受理証明書を受信
S62'
イベント受理証明書、時刻証明書を送信
S50'
イベント受理証明書を送信

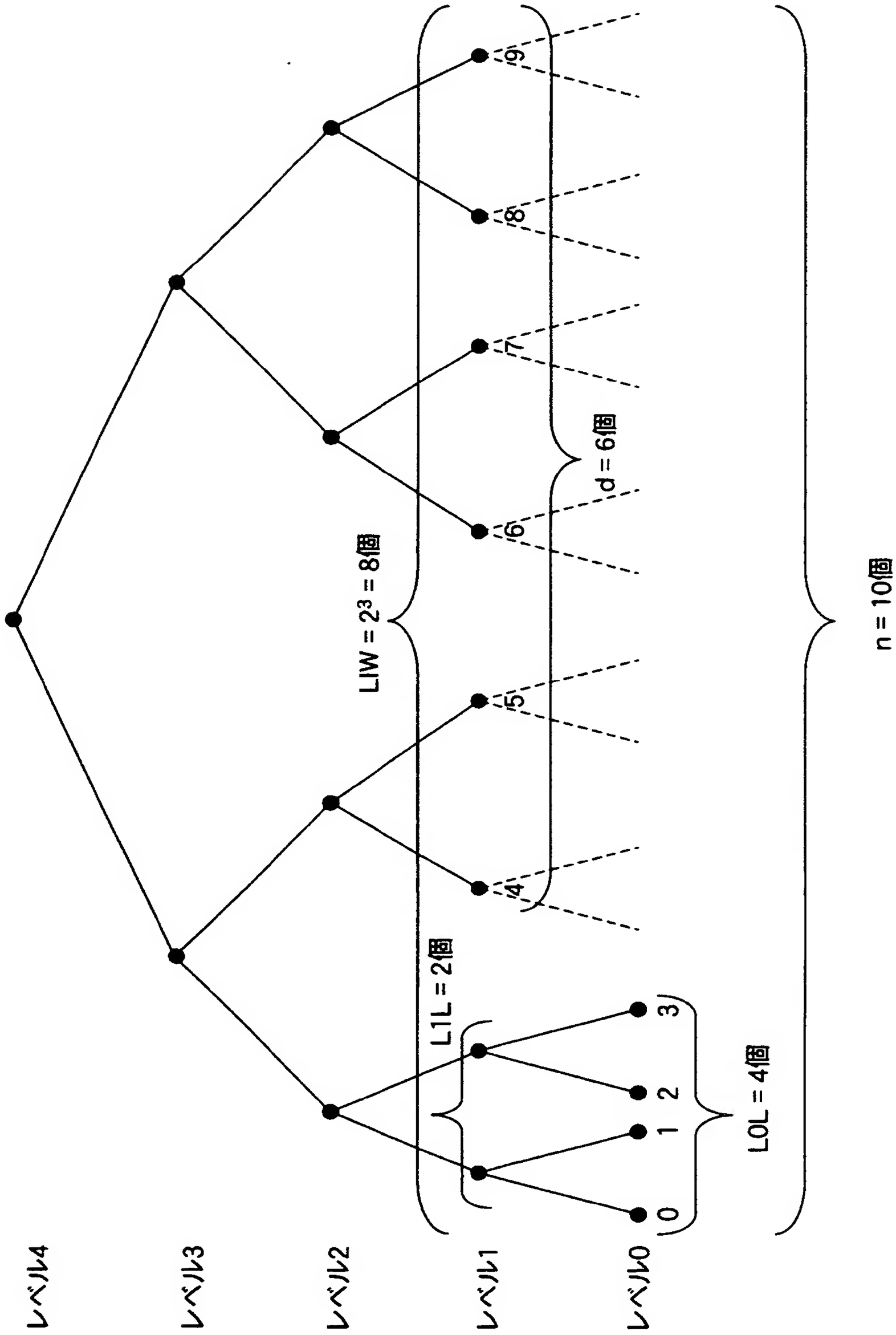
[図24]


利用者装置(時刻証明装置)10j
監査装置20i
S120'
S121
監査用受理証明書を受信
S122
時刻情報を取得
S123
時刻付き監査用受理証明書を作成
証明装置1
S110'
監査用受理証明書を送信

[図25]


利用者装置(時刻証明装置)10j
監査装置20i
S510'
S511'
監査OK
S512'
区間時刻証明証を発行
S513'
区間時刻証明証を監査結果に含める
証明装置1

[図26]


レベル4
レベル3
レベル2
レベル1
レベル0
LIW = 2³ = 8個
L1L = 2個
L0L = 4個
d = 6個
n = 10個
0
1
2
3
4
5
6
7
8
9

[図27]

(1) ループ1：構成法その3では、定められた時間間隔が終了するまで
以下の処理を繰り返す.
(1.1) 受け付けた要求をxに置く.
(1.2) nを1インクリメントする.
(1.3) ループ2：j = 0, .., Kに対して以下の処理を行う.
(1.3.1) i : = i_jと置く.
(1.3.2) j = 0のときは、A_j[i] : = x
(即ち、ノード(j, i_j)にxを設定する).
(1.3.3) j ＞ 0のときは、以下を行う.
・ x0 : = A_{j-1}[index(leftChiild(j, i))]
(ノード(j, i) のleft-childに割り付けられた値をx0とおく).
・ x1 : = A_{j-1}[index(rightChild(j, i))]
(ノード(j, i) のright-childに割り付けられた値をx1とおく).
・ x2 : = h(x0 || x1)を計算する.
・ A_j[i] : = x2
(即ち、ノード(j, i) にx2を割り付ける).
(1.3.4) i_j を1インクリメントする.
(1.3.5) iが偶数のときは、ループ2を抜ける.
ループ2終了.
ループ1終了.

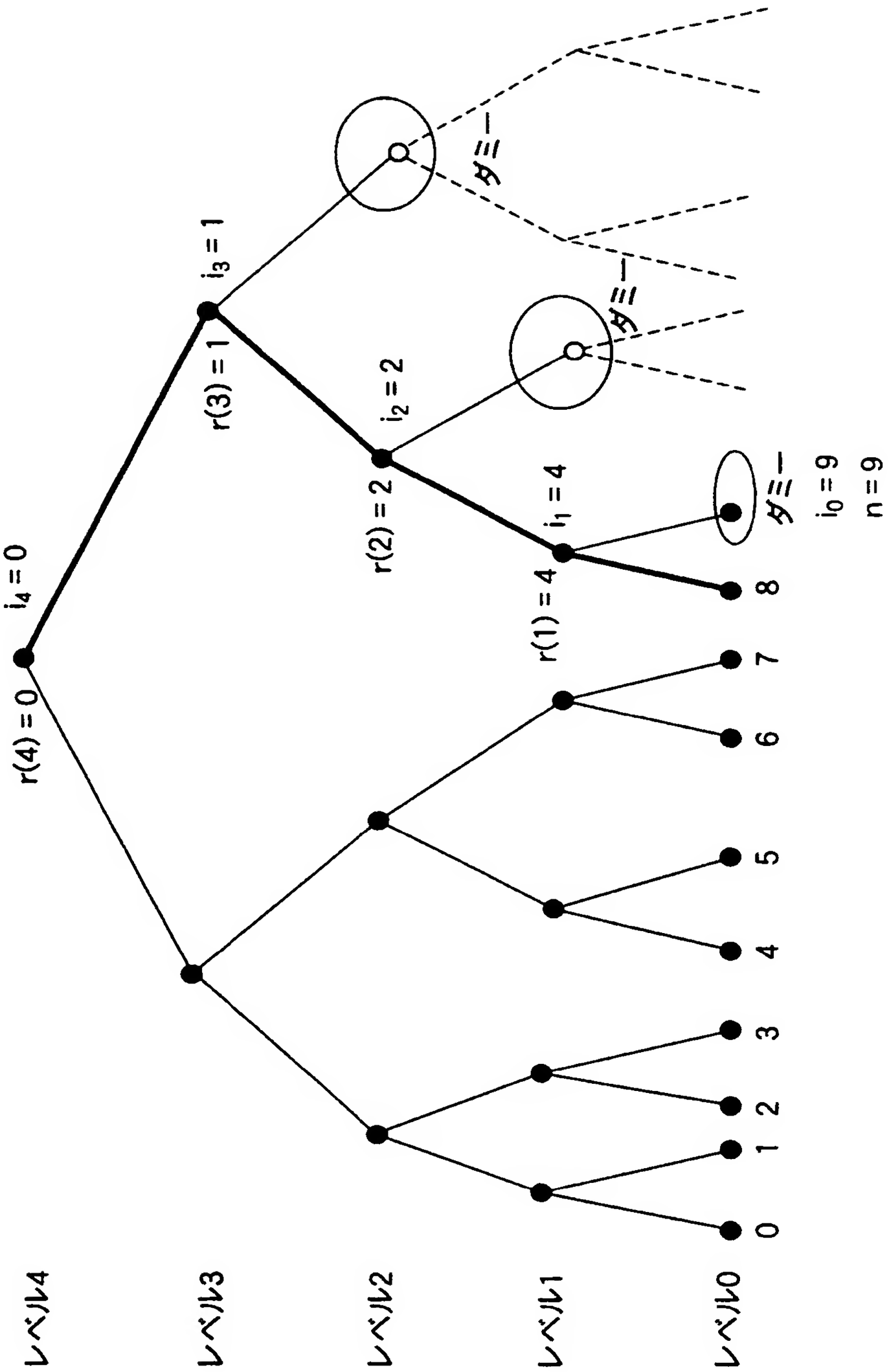
処理手順1

[図28]

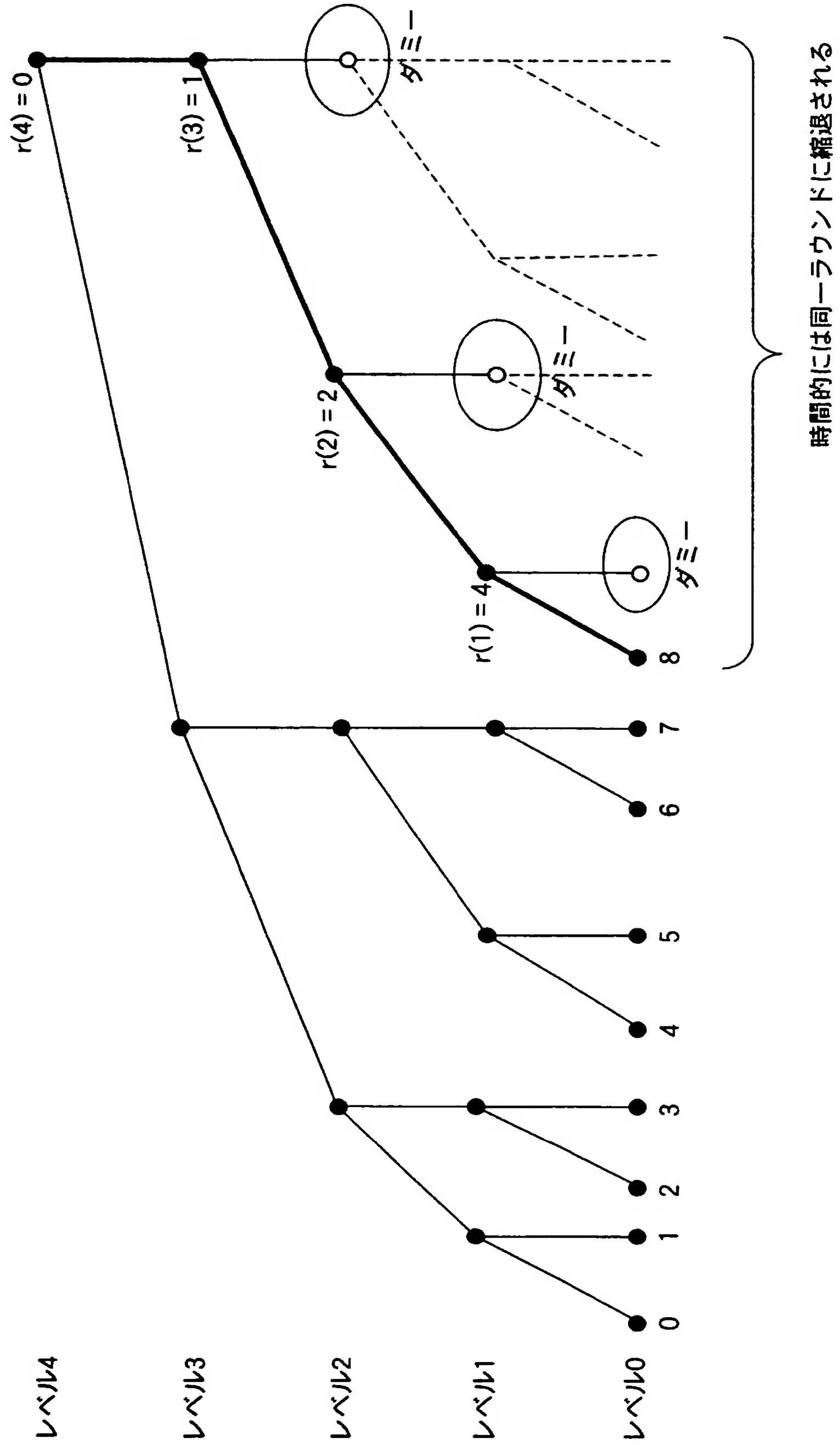
(2)終了時間がきて、ループ1を抜けた後では次の処理を行う.
(2.1) k : = ceiling($\log_2$(n))とおく.
(2.2) rtPath(k,0, n−1) を計算し、その結果を((0, r(0)), ..., (k,r(k)))と置く.
(2.3) ループ3 : j = 0, ..., kに対して以下の処理を行う.
(2.3.1) i = i_j とおく.
(2.3.2) j = 0のとき :
(2.3.2.1) i が奇数のとき :
・ダミー値 r : = R(0,i) を生成する.
・A_j[i] : = r
(ノード (0, i) にrを割り当て).
・b_j : = trueと置く.
・i_j を1インクリメントする.
(2.3.2) 0 <j ≦ kのとき :
(2.3.2.1) i = r(j) のとき :
(即ち、ノード(j, i) が rtPath(k, 0, n−1) 上にあるとき):
(2.3.2.1.1) x0 : = A_{j-1}[index(leftChiild(j, i))]
(ノード(j, i) のleft−childに割り付けられた値をx0と置く).
(2.3.2.1.2) x1 : = A_{j-1}[index(rightChild(j, i))]
(ノード(j, i) のright−childに割り付けられた値をx1と置く).
(2.3.2.1.3) x2 : = h(x0 ‖ x1)を計算する
(2.3.2.1.4) A[i] : = x2
(即ち、ノード(j, i) にx2を割り付ける).
(2.3.2.1.5) iが偶数で j<k のとき :
・ i を1インクリメントする.
・ r : = R(j, i)を計算し、A_j[i] : = r
(即ち、ノード(j, i) にrを割り当る).
・ b_j : = trueと置く.
・ i_j : = i+1と置く.
(2.3.2.2) i = r(j)+1で i が奇数で j<k のとき :
・ r : = R(j, i)を計算し、A_j[i] : = r
(即ち、ノード(j, i) にrを割り当る).
・ b_j : = trueと置く.
・ i_jを1インクリメントする.
ループ3終了.

処理手順2

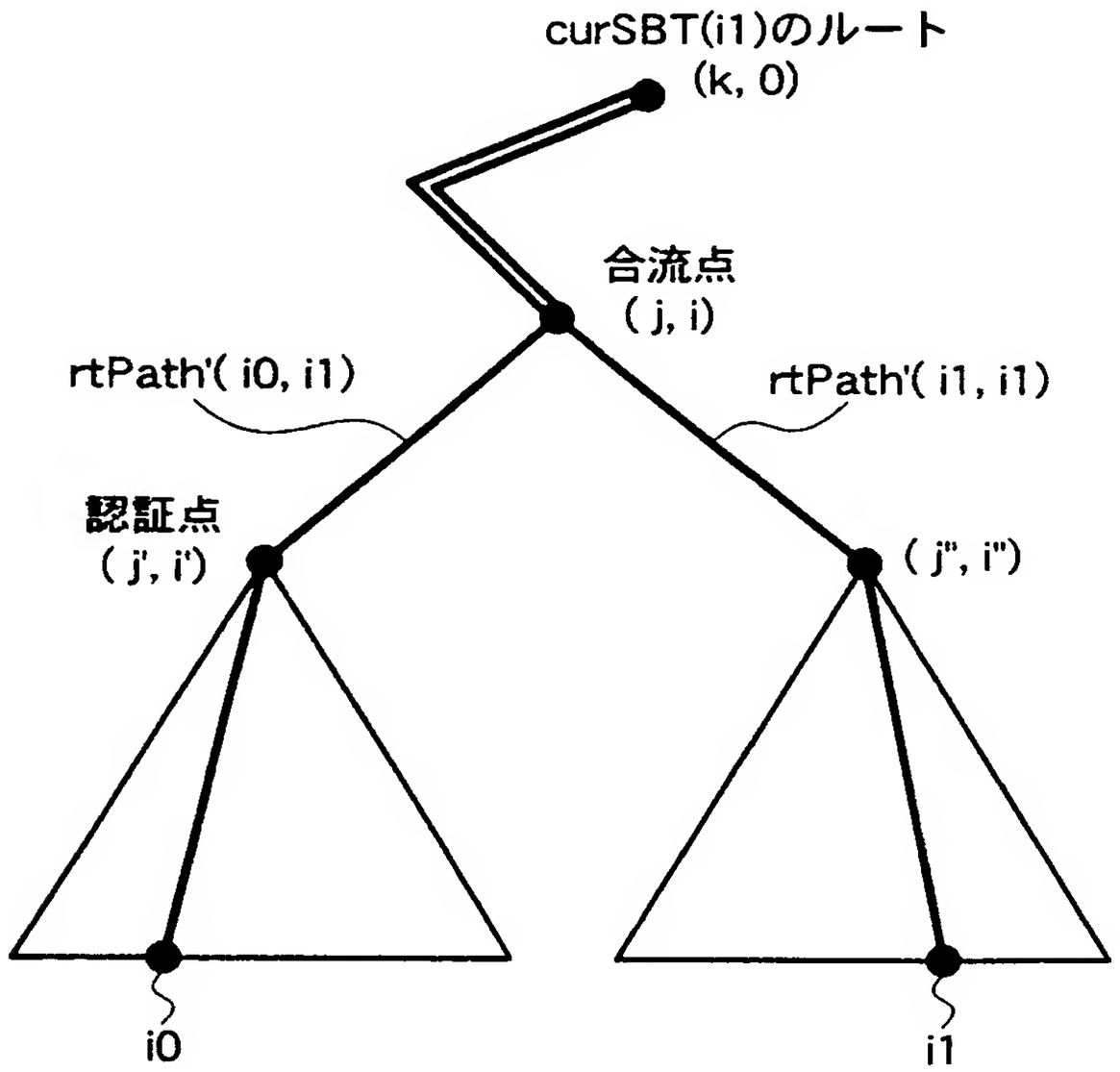
[図29]


$i_3 = 1$
$i_2 = 2$
$i_1 = 4$
$i_0 = 9$
$n = 9$
$i_4 = 0$
r(4) = 0
r(3) = 1
r(2) = 2
r(1) = 4
0
1
2
3
4
5
6
7
8
レベル4
レベル3
レベル2
レベル1
レベル0

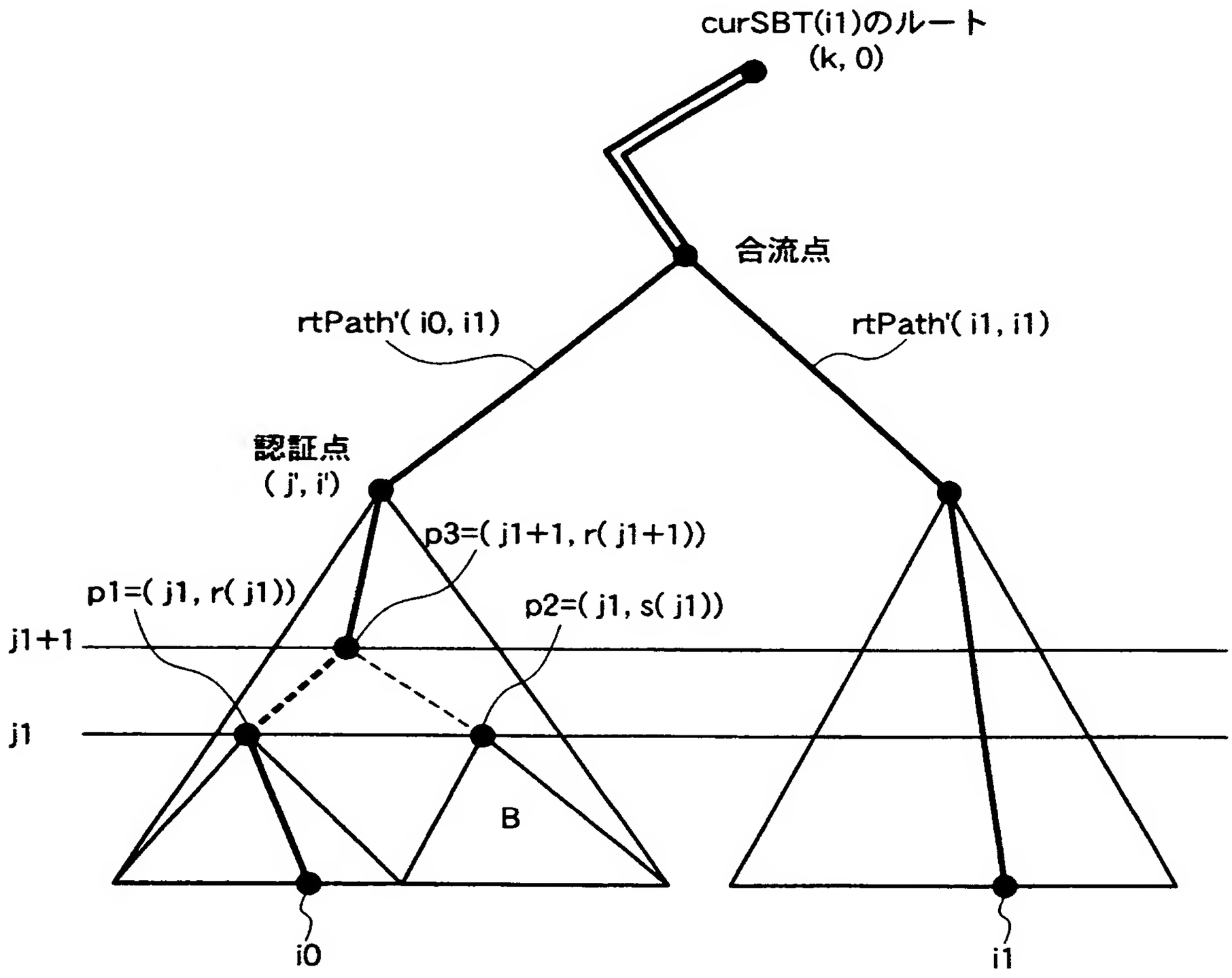
[図30]


r(4) = 0
r(3) = 1
r(2) = 2
r(1) = 4
8
7
6
5
4
3
2
1
0
レベル4
レベル3
レベル2
レベル1
レベル0
ダミー
ダミー
ダミー
時間的には同一ラウンドに縮退される

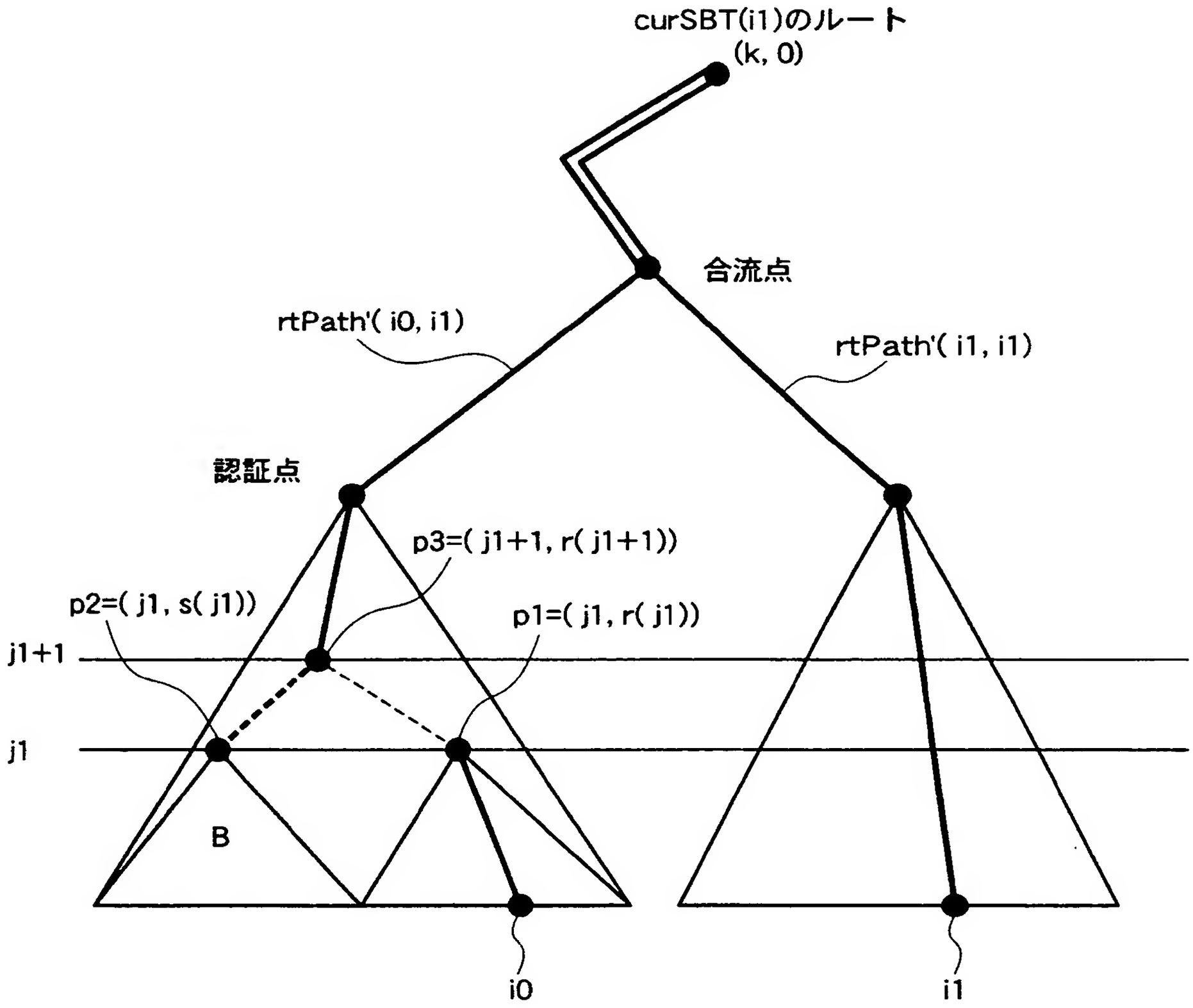
[図31]


curSBT(i1)のルート
(k, 0)
合流点
(j, i)
rtPath'(i0, i1)
rtPath'(i1, i1)
認証点
(j', i')
(j", i")
i0
i1

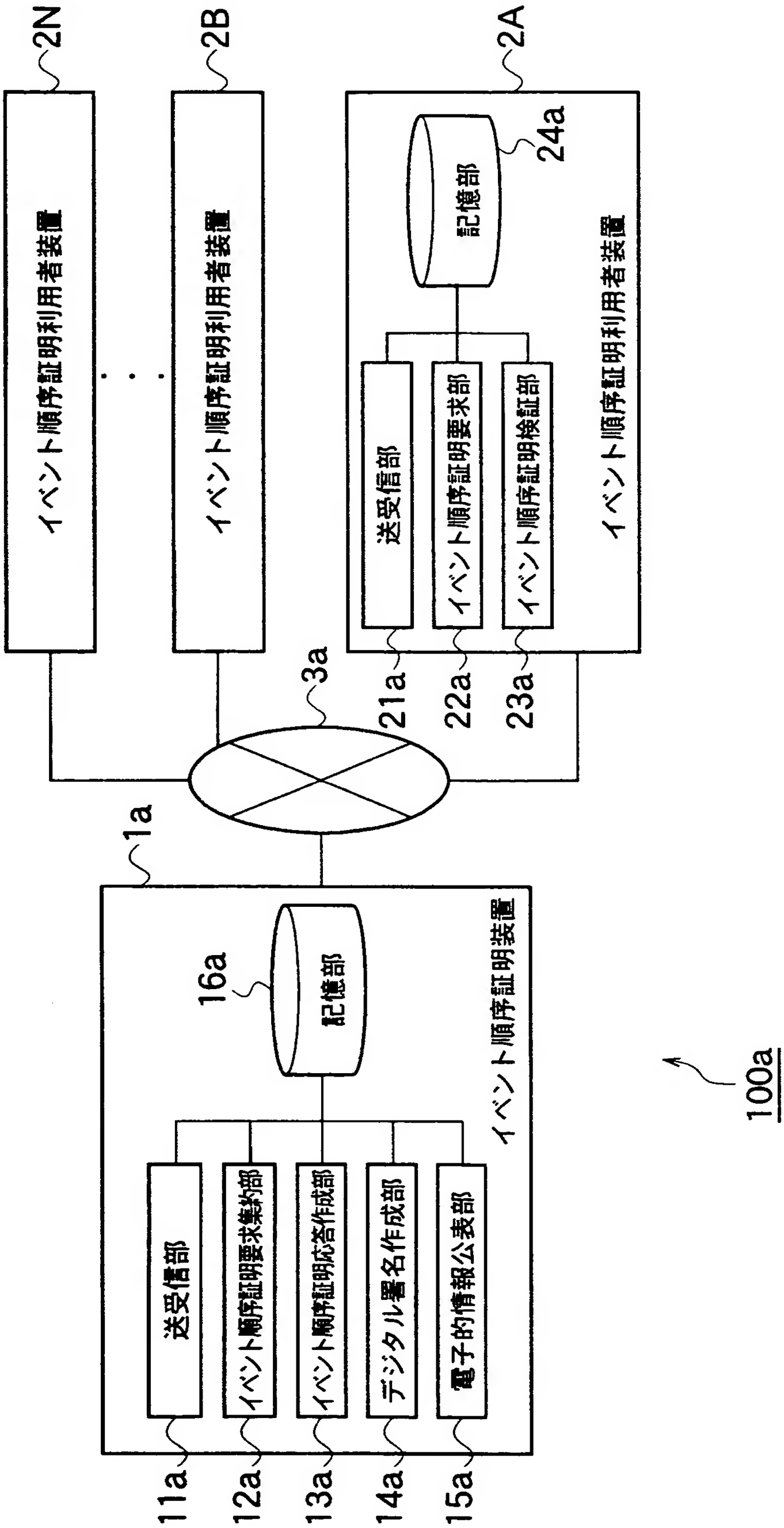
[図32]


curSBT(i1)のルート
(k, 0)
合流点
rtPath'(i0, i1)
rtPath'(i1, i1)
認証点
(j', i')
p3=(j1+1, r(j1+1))
p1=(j1, r(j1))
p2=(j1, s(j1))
j1+1
j1
B
i0
i1

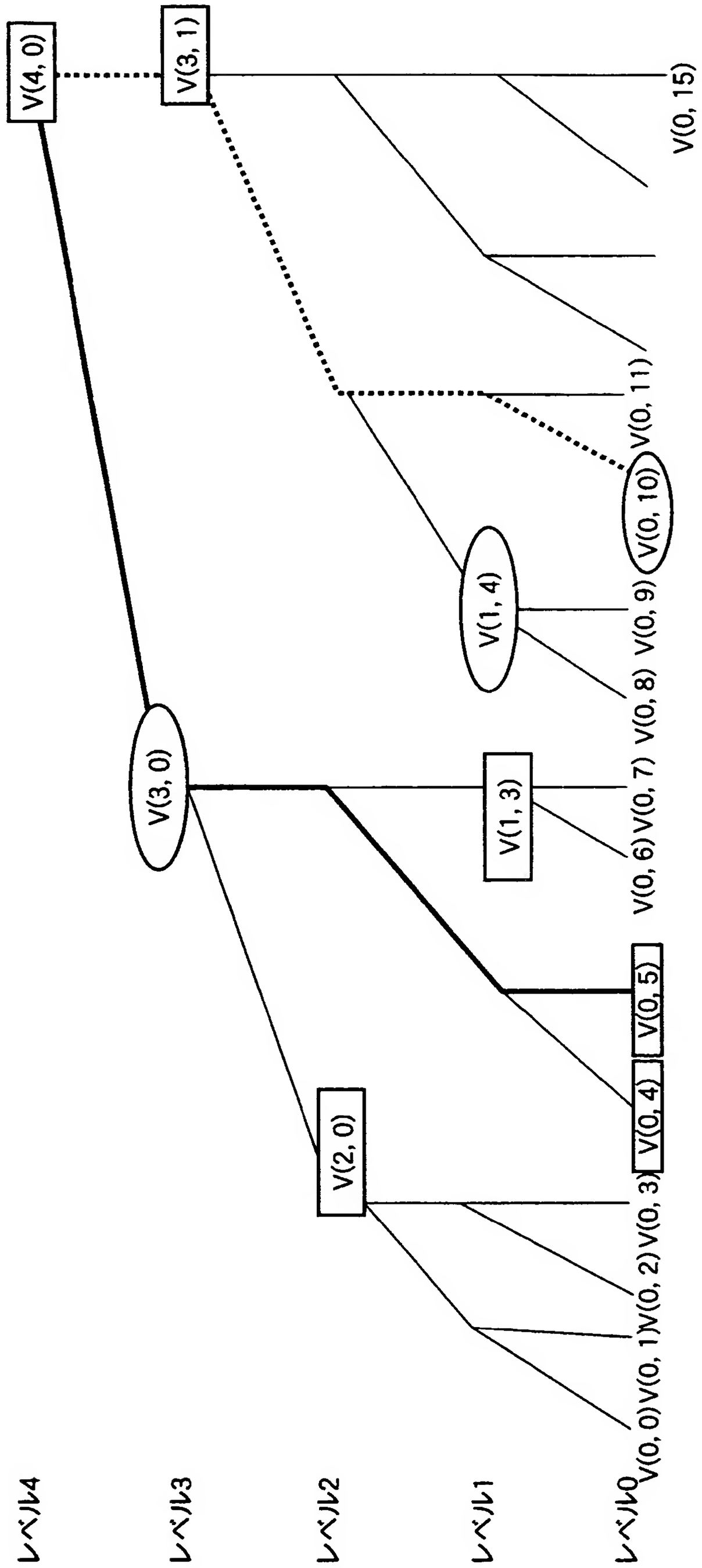
[図33]


curSBT(i1)のルート
(k, 0)
合流点
rtPath'(i0, i1)
rtPath'(i1, i1)
認証点
p3=(j1+1, r(j1+1))
p2=(j1, s(j1))
p1=(j1, r(j1))
j1+1
j1
B
i0
i1

[図34]


100a
イベント順序証明装置
送受信部
イベント順序証明要求集約部
イベント順序証明応答作成部
デジタル署名作成部
電子的情報公表部
記憶部
11a
12a
13a
14a
15a
16a
1a
3a
イベント順序証明利用者装置
送受信部
イベント順序証明要求部
イベント順序証明検証部
記憶部
21a
22a
23a
24a
2A
2B
2N

[図35]

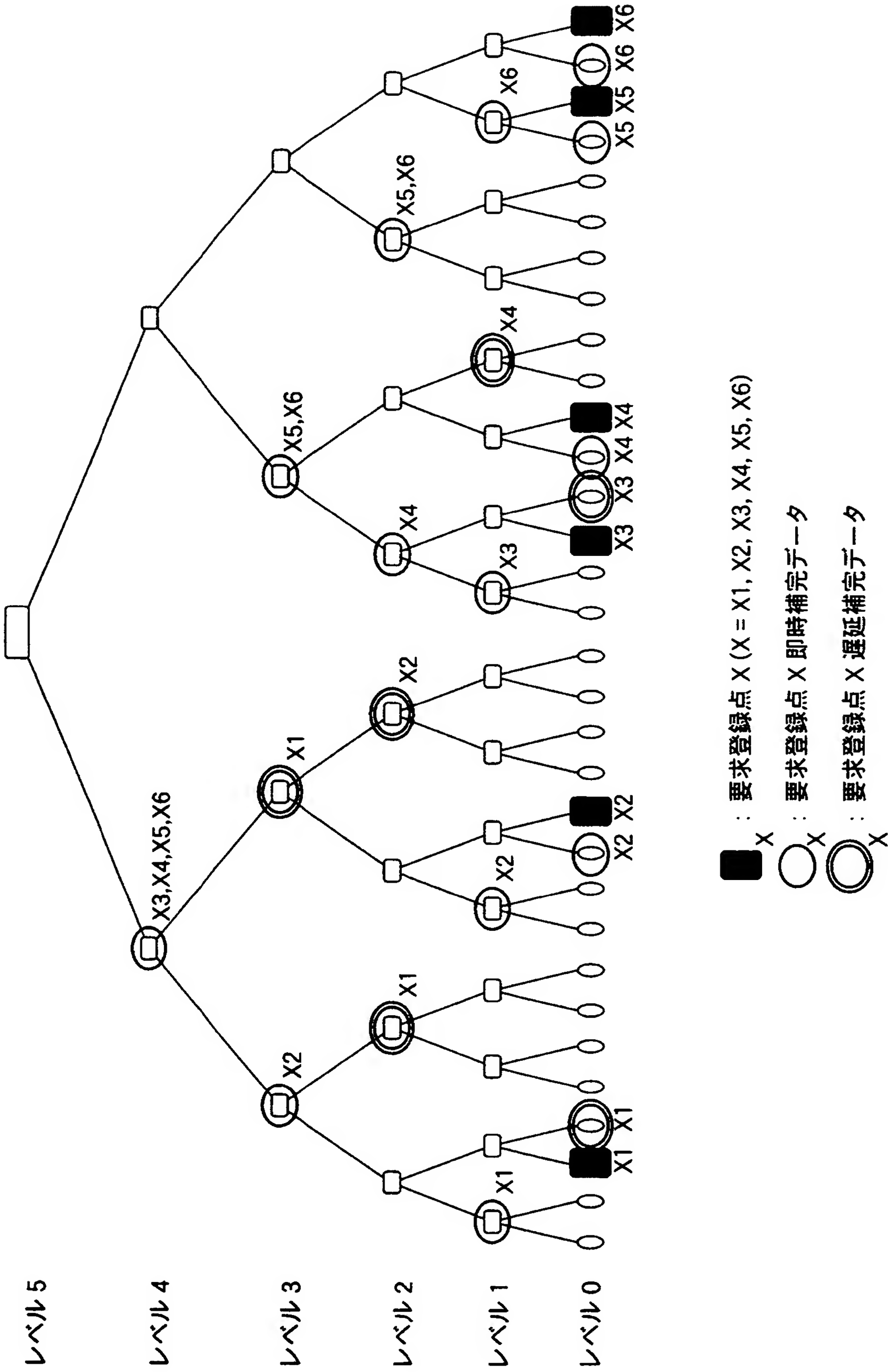

V(4, 0)
V(3, 1)
V(0, 15)
V(0, 11)
V(0, 10)
V(0, 9)
V(0, 8)
V(0, 7)
V(0, 6)
V(0, 5)
V(0, 4)
V(0, 3)
V(0, 2)
V(0, 1)
V(0, 0)
V(1, 4)
V(3, 0)
V(1, 3)
V(2, 0)
レベル4
レベル3
レベル2
レベル1
レベル0

[図36]

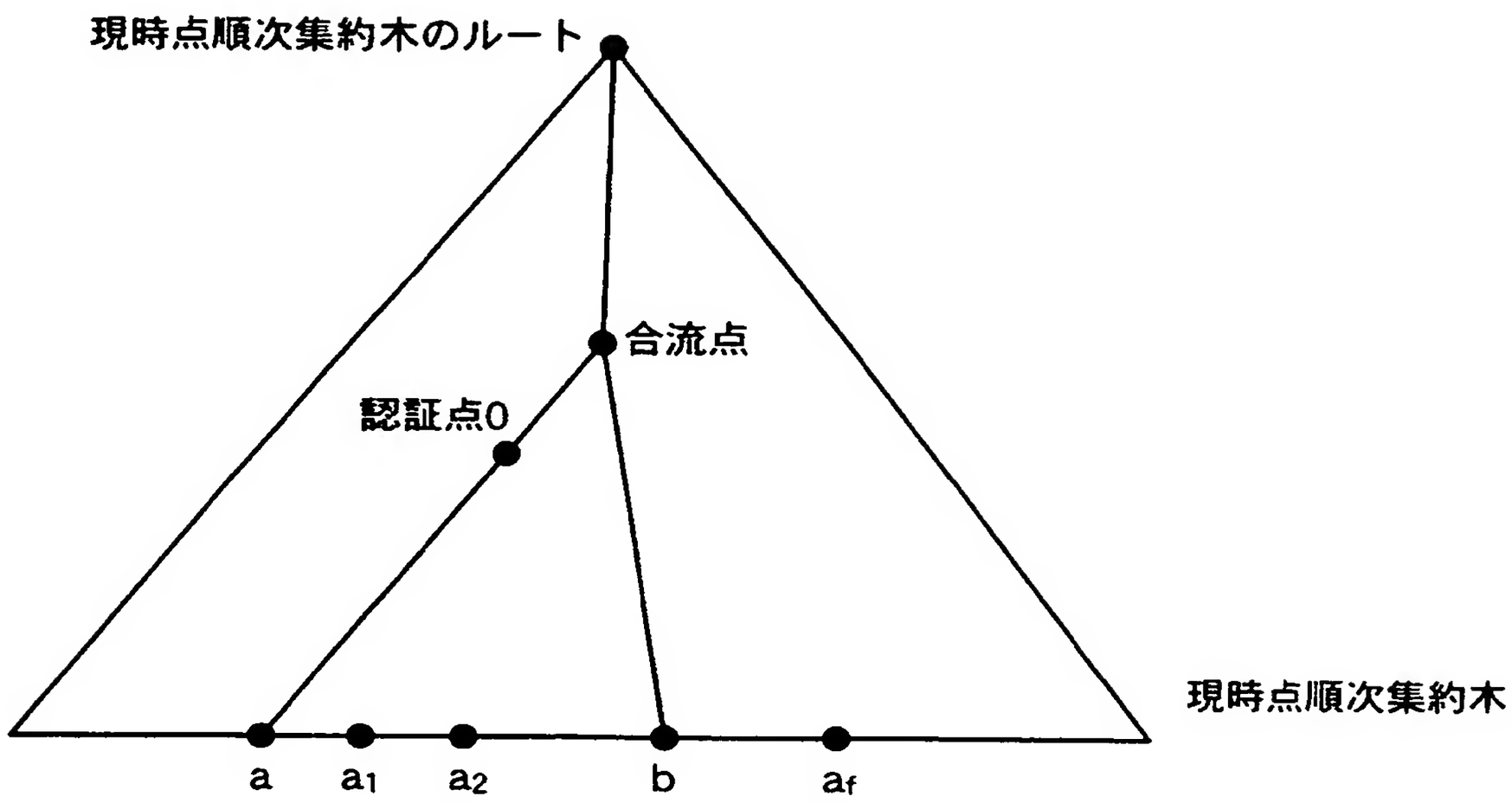
イベント順序受理証明書
EOC(y)

項目	記号	必須
元デジタルデータ	y	○
順次割当データ	z	○
順次集約木番号	n	○
順次集約木リーフ番号	i	○
登録点の即時補完データ(位置情報、割当値)	SK	○
過去の各登録点の遅延補完データ(位置情報、割当値)	TK	○

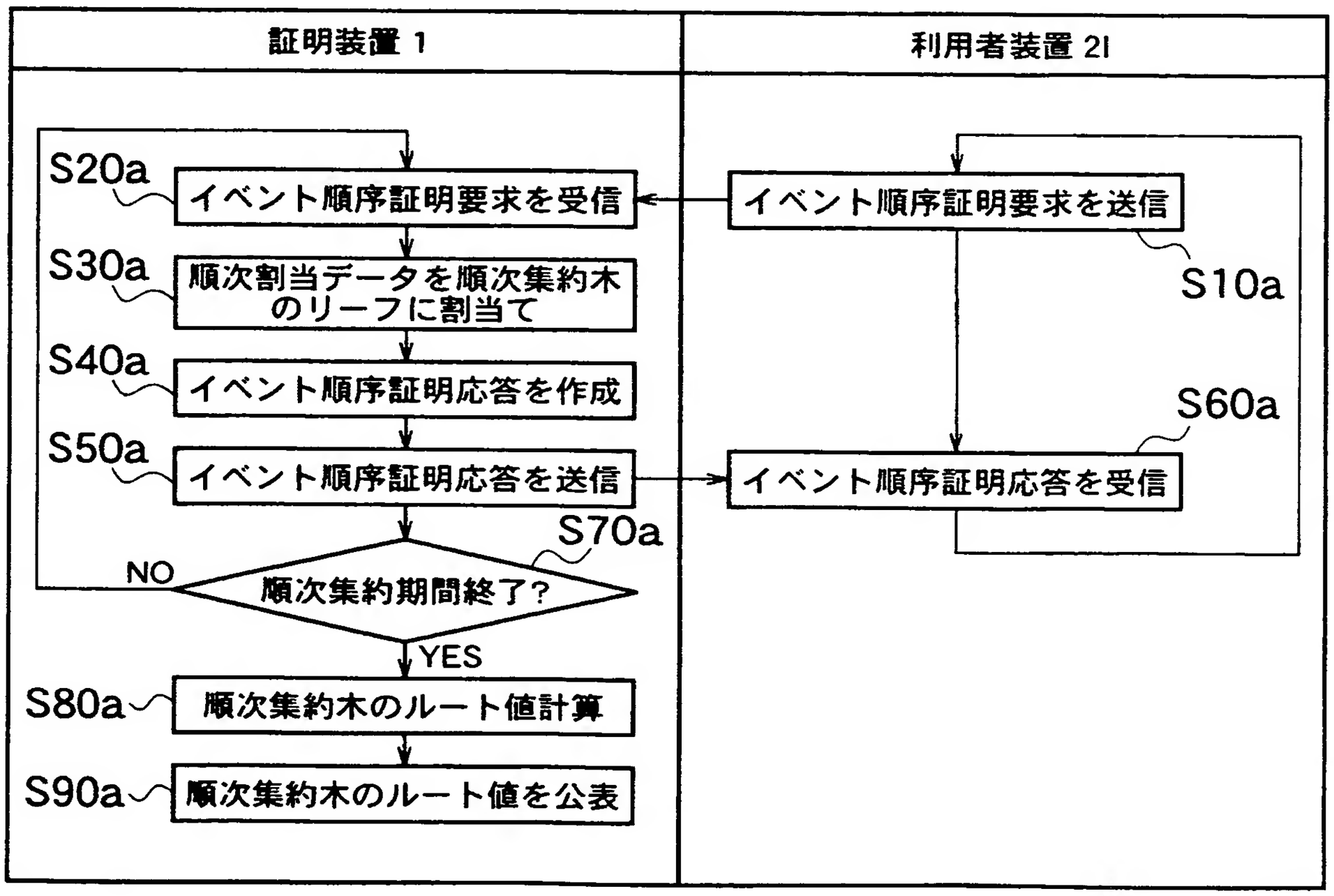
[図37]


レベル5
レベル4
レベル3
レベル2
レベル1
レベル0
X3,X4,X5,X6
X2
X1
X5,X6
X4
X3
X2
X1
X6
X5
X6 X6
X5 X5
X5,X6
X4
X4
X3
X3 X4 X4
X3
X2
X2 X2
X1
X1 X1
X1
■ X : 要求登録点 X (X = X1, X2, X3, X4, X5, X6)
○ X : 要求登録点 X 即時補完データ
◎ X : 要求登録点 X 遅延補完データ

[図38]


現時点順次集約木のルート
合流点
認証点0
a
a1
a2
b
af
現時点順次集約木

[図39]


証明装置 1
利用者装置 2l
S20a
イベント順序証明要求を受信
S30a
順次割当データを順次集約木のリーフに割当て
S40a
イベント順序証明応答を作成
S50a
イベント順序証明応答を送信
S70a
NO
順次集約期間終了?
YES
S80a
順次集約木のルート値計算
S90a
順次集約木のルート値を公表
イベント順序証明要求を送信
S10a
S60a
イベント順序証明応答を受信